An Overture Overview

Bill Henshaw

Centre for Applied Scientific Computing,
Lawrence Livermore National Laboratory,
Livermore, CA, USA  94551

`www.llnl.gov/casc/Overture`

Presented at NASA Ames, June 2007.

Acknowledgments

Supported by

Department of Energy, Office of Science

MICS Program: Mathematical, Information, and Computational Sciences

SciDAC: Scientific Discovery through Advanced Computing

LLNL: Laboratory Directed Research and Development (LDRD) program

Current Overture developers

Kyle Chand

Bill Henshaw

Collaborators

Don Schwendeman (RPI),

Tom Hagstrom (UNM),

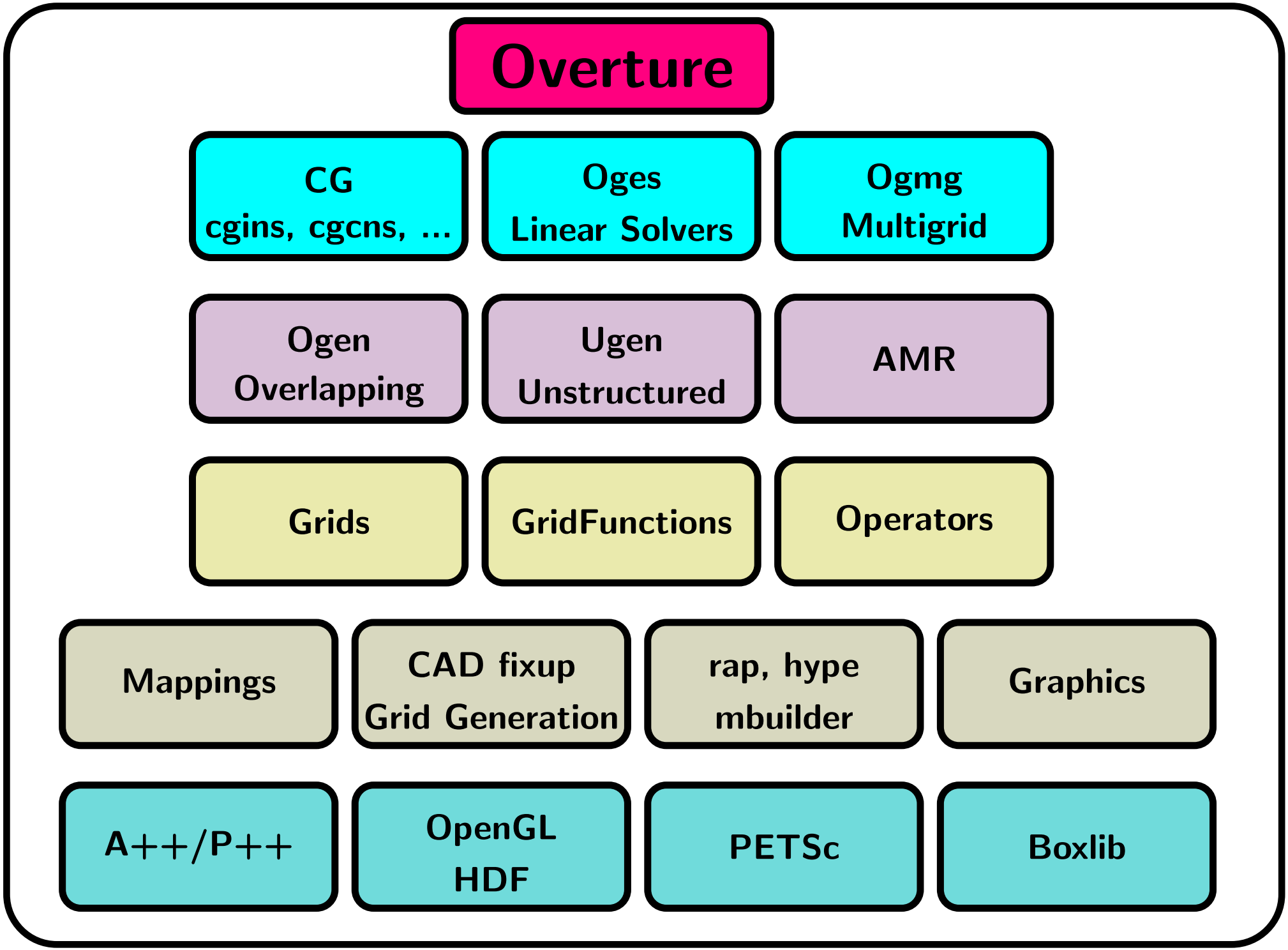
Nikos Nikiforakis (U. Cambridge),

Jeff Banks (SNL)

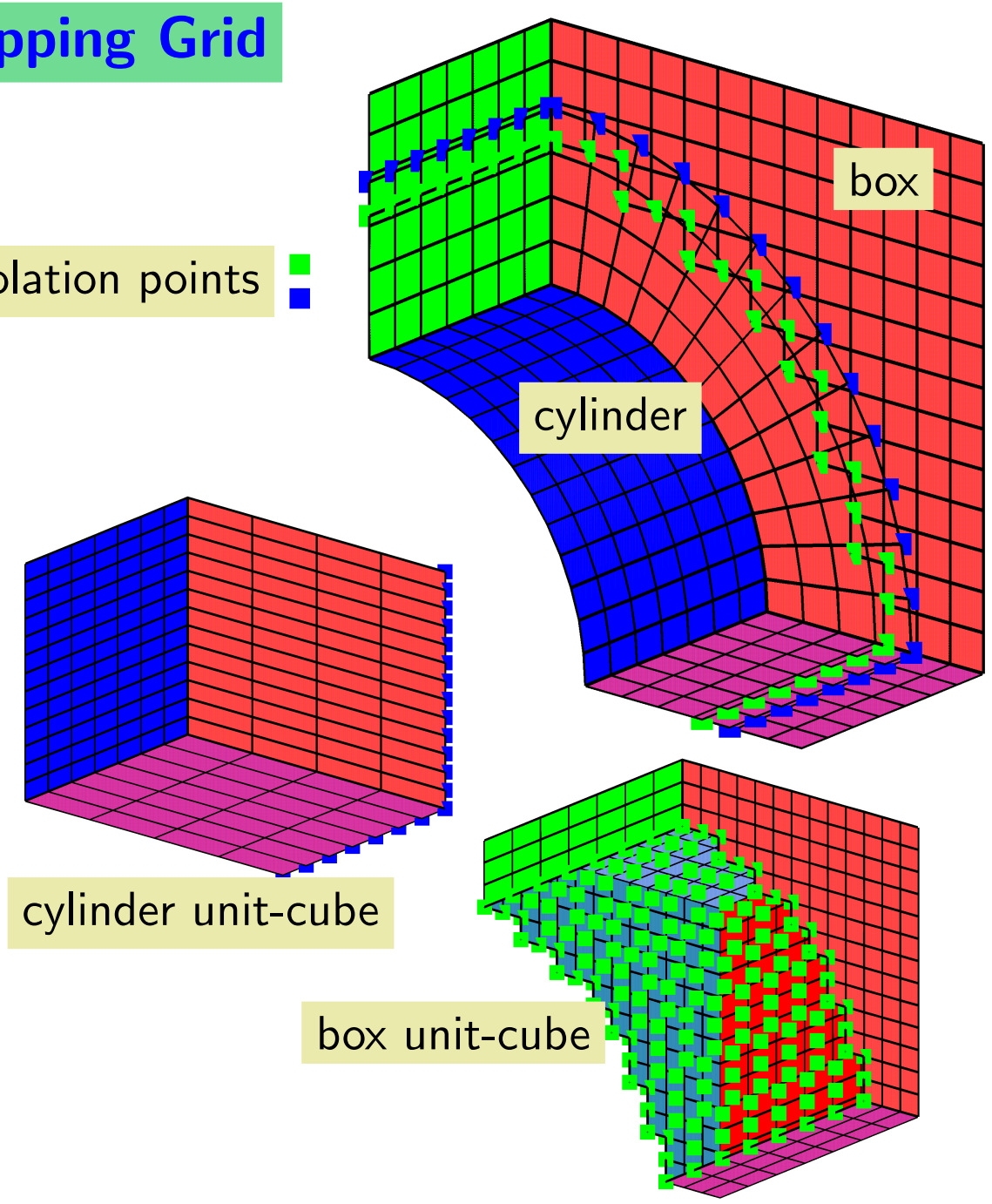
Overture is toolkit for solving partial differential equations on structured, overlapping and hybrid grids.

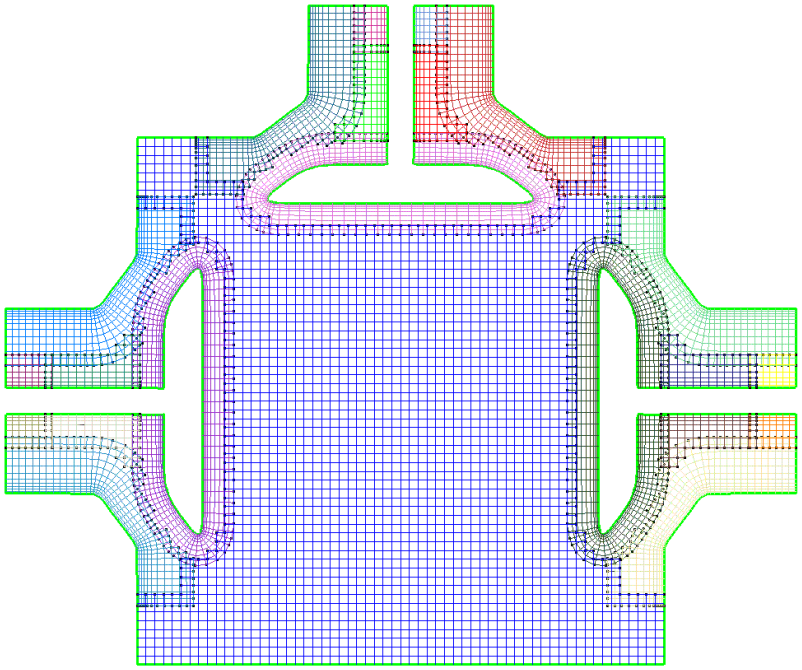
Key features:

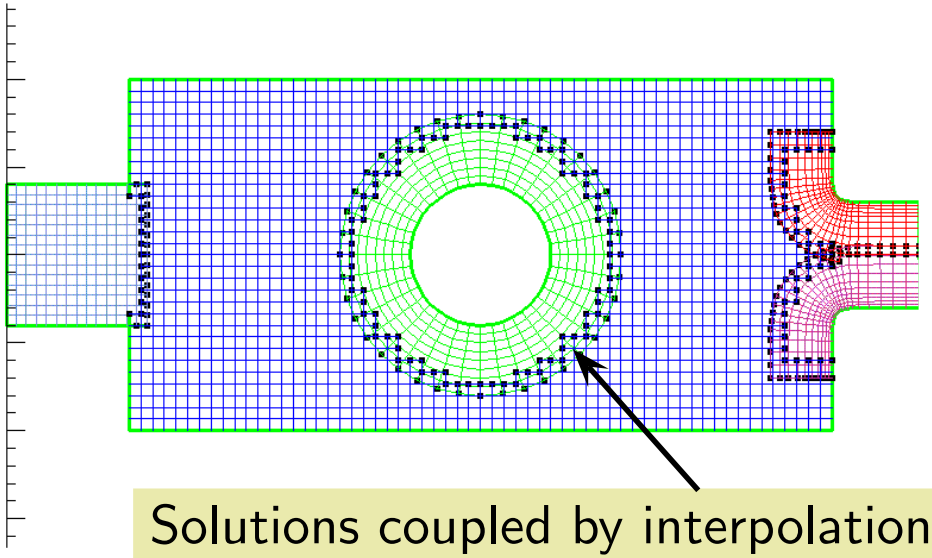
- provides a high level C++ interface for rapid prototyping of PDE solvers.
- built upon optimized C and fortran kernels.
- provides a library of finite-difference operators: conservative and non-conservative, 2nd, 4th, 6th and 8th order accurate approximations.
- support for moving grids.
- support for block structured adaptive mesh refinement (AMR).
- extensive grid generation capabilities.
- CAD fixup tools (for CAD from IGES files).
- interactive graphics and data base support (HDF).
- PDE solvers built upon Overture include:
  - cgins: incompressible Navier-Stokes with heat transfer.
  - cgcns: compressible Navier-Stokes, reactive Euler equations.
  - cgmx: time domain Maxwell's equations solver.


Overture
CG
cgins, cgcns, ...
Oges
Linear Solvers
Ogmg
Multigrid
Ogen
Overlapping
Ugen
Unstructured
AMR
Grids
GridFunctions
Operators
Mappings
CAD fixup
Grid Generation
rap, hype
mbuilder
Graphics
A++/P++
OpenGL
HDF
PETSc
Boxlib

3D Overlapping Grid

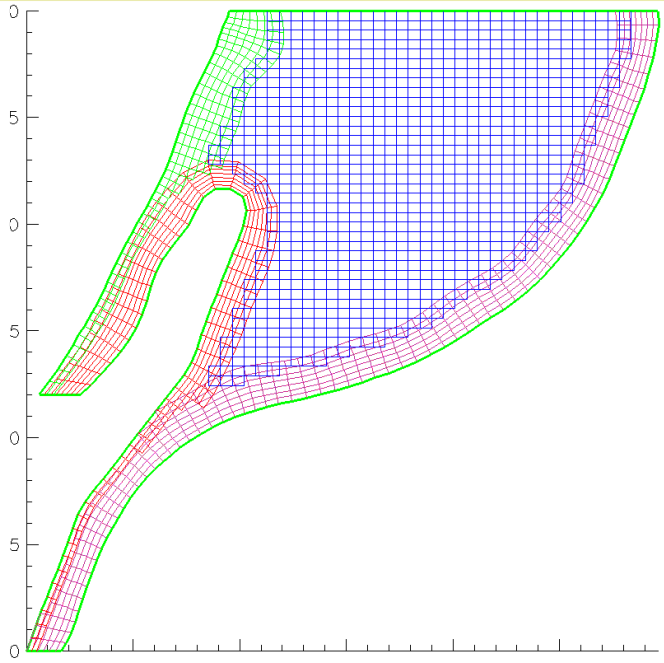


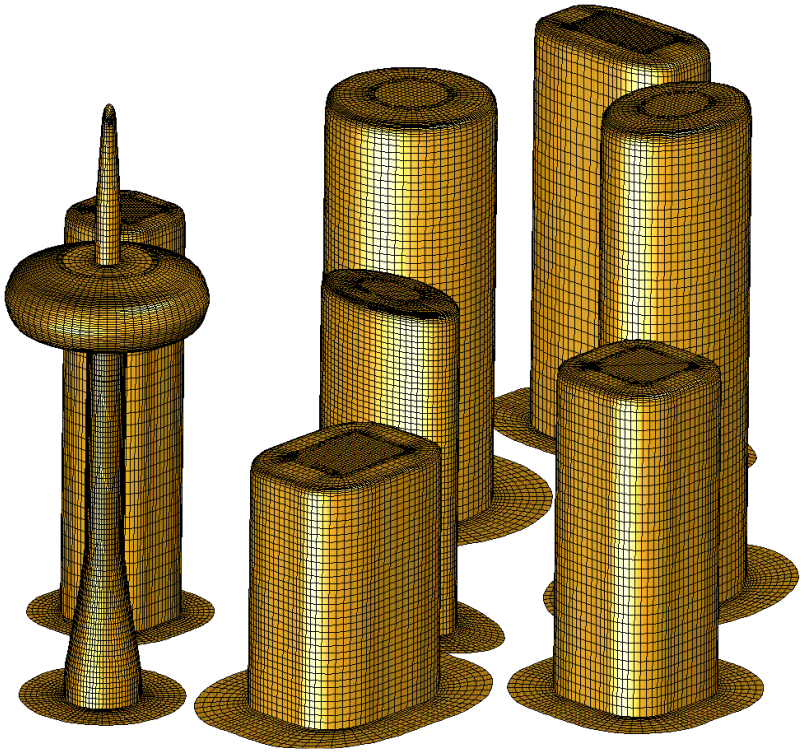


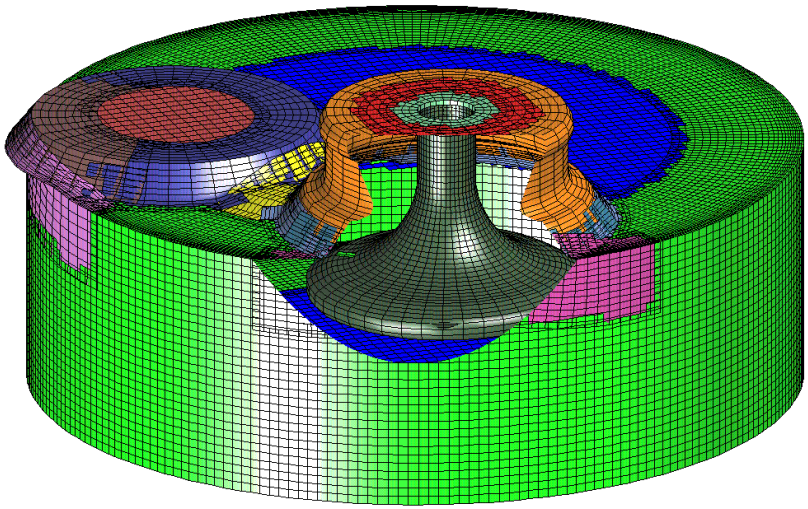


Solutions coupled by interpolation

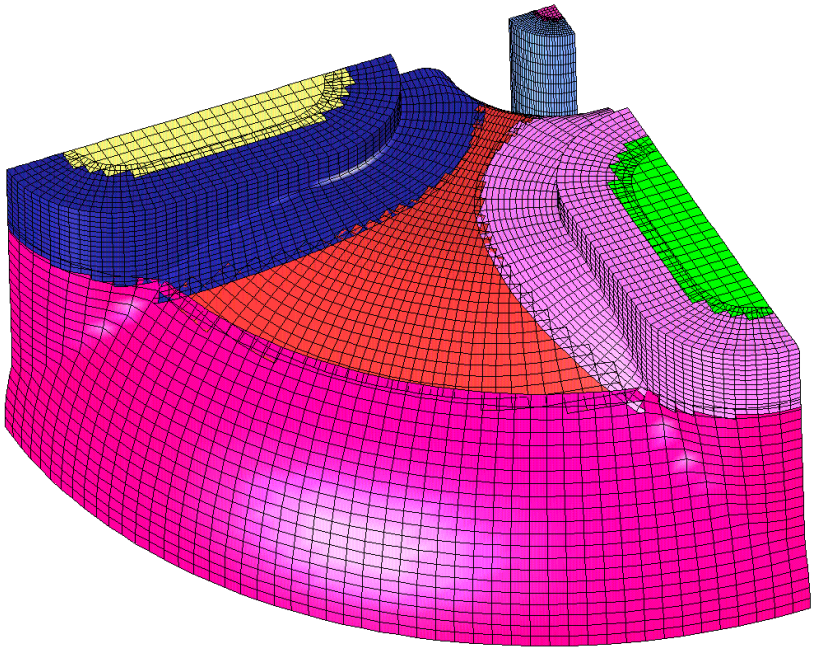
Sample 2D overlapping grids

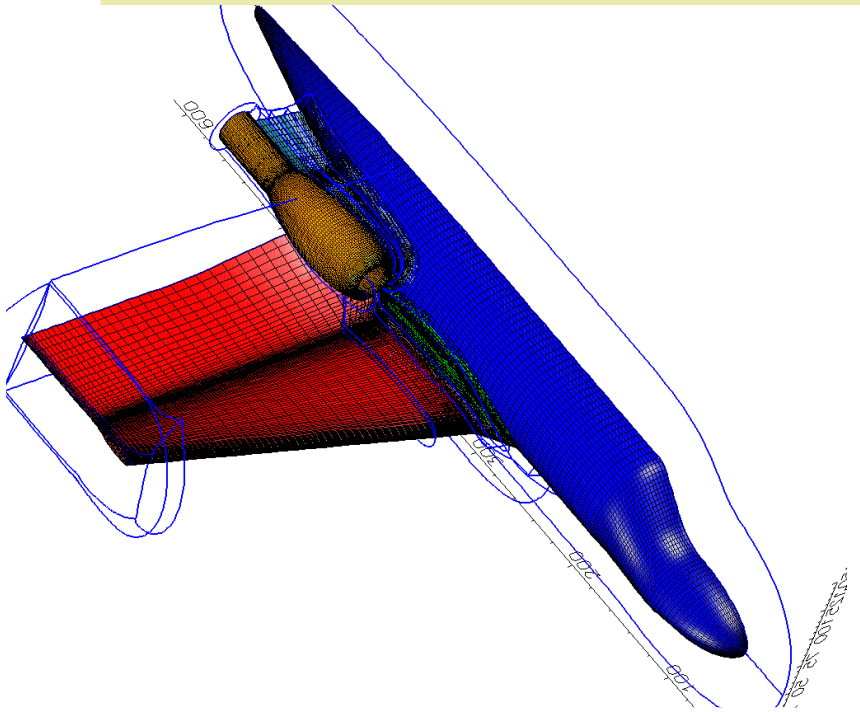


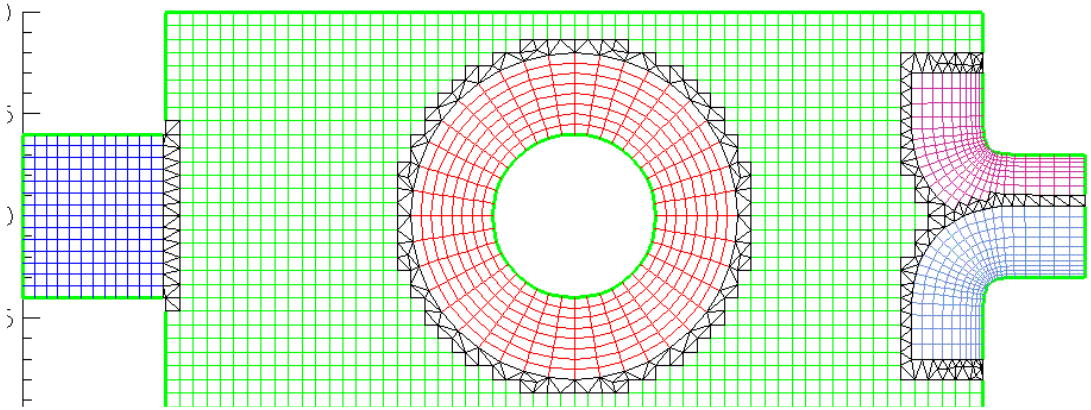


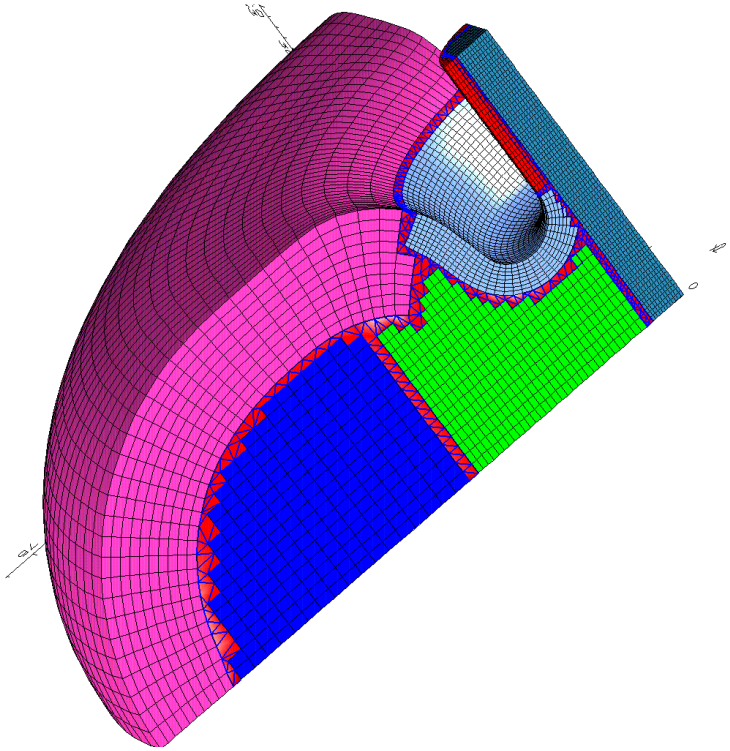


Sample 3D overlapping grids

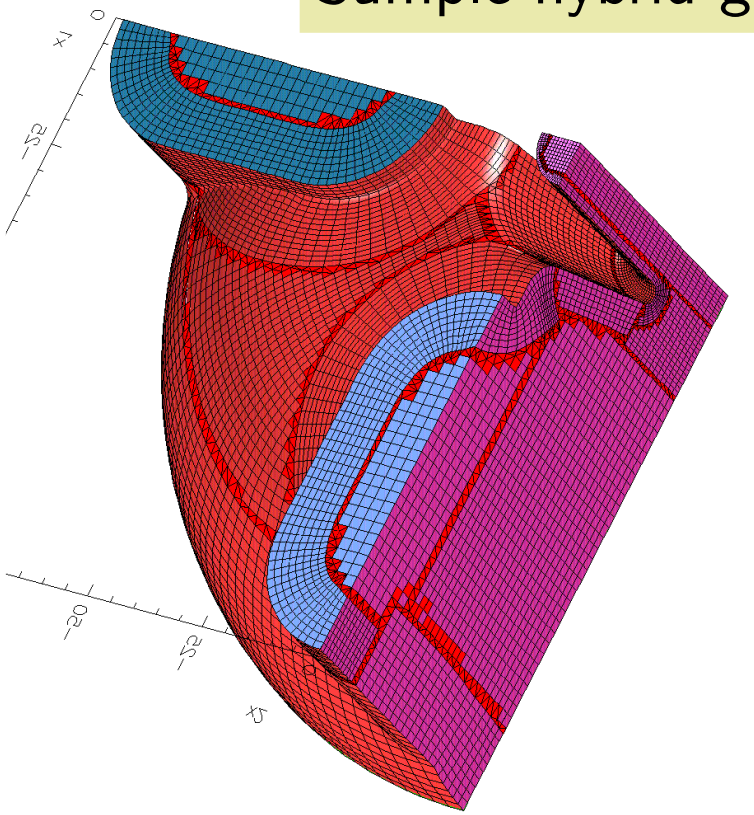


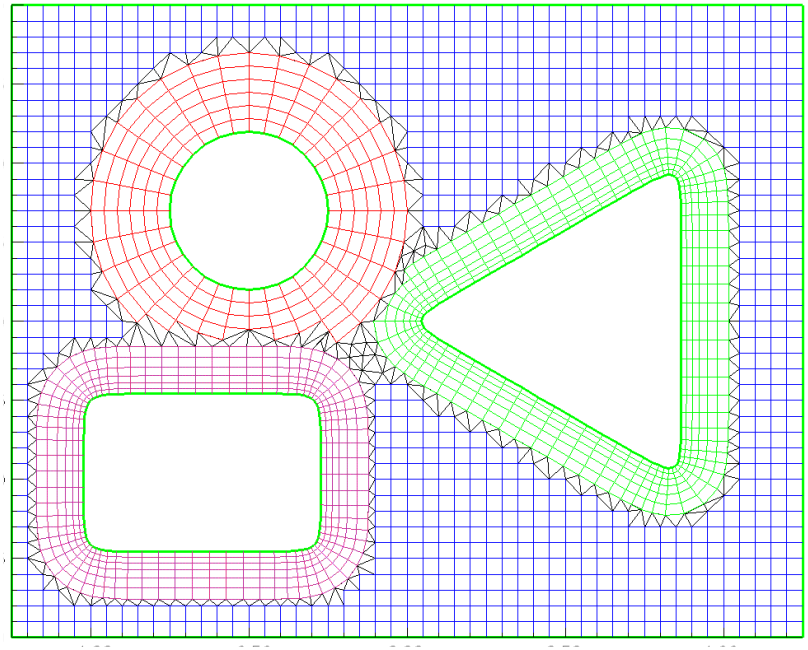


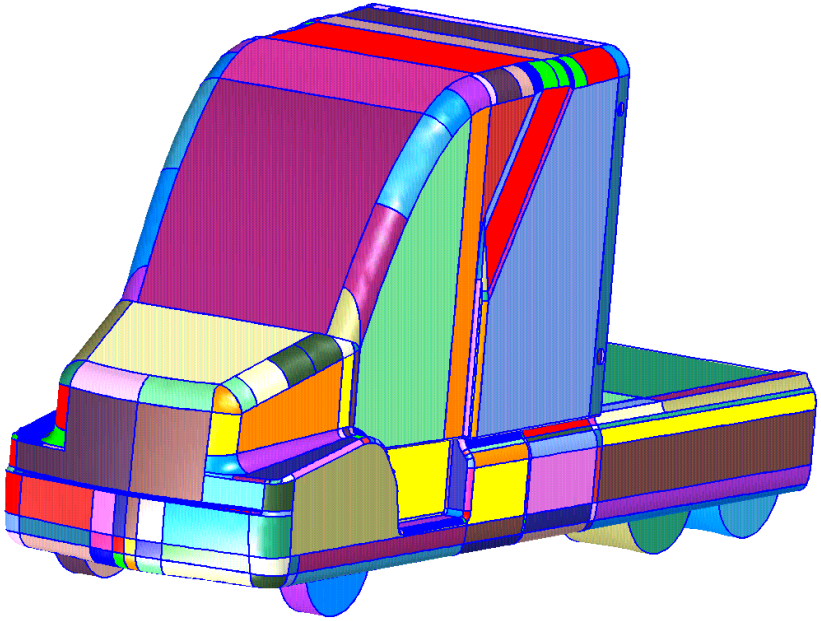




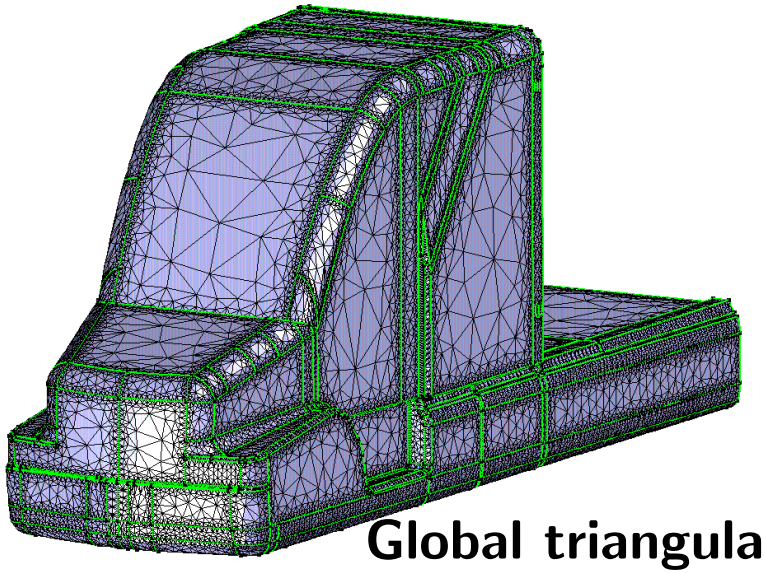
Sample hybrid grids



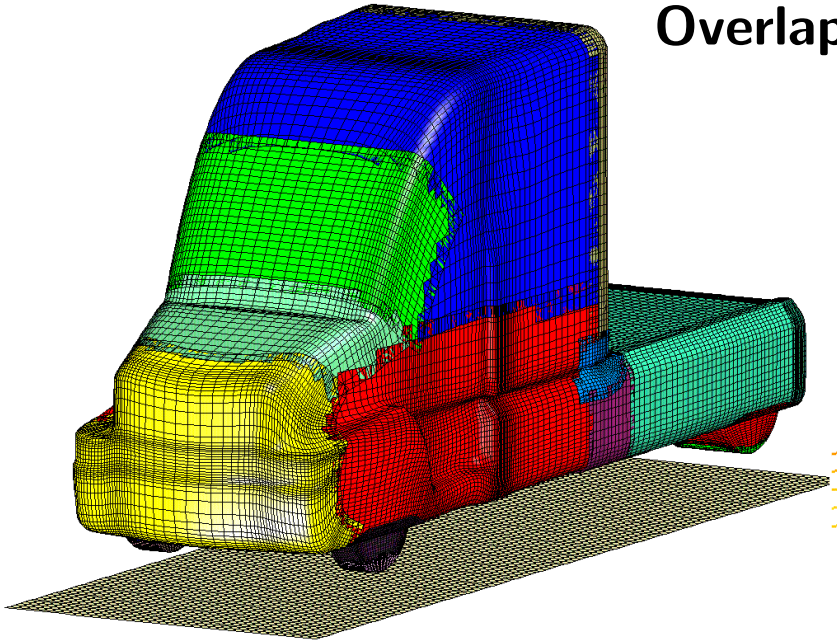




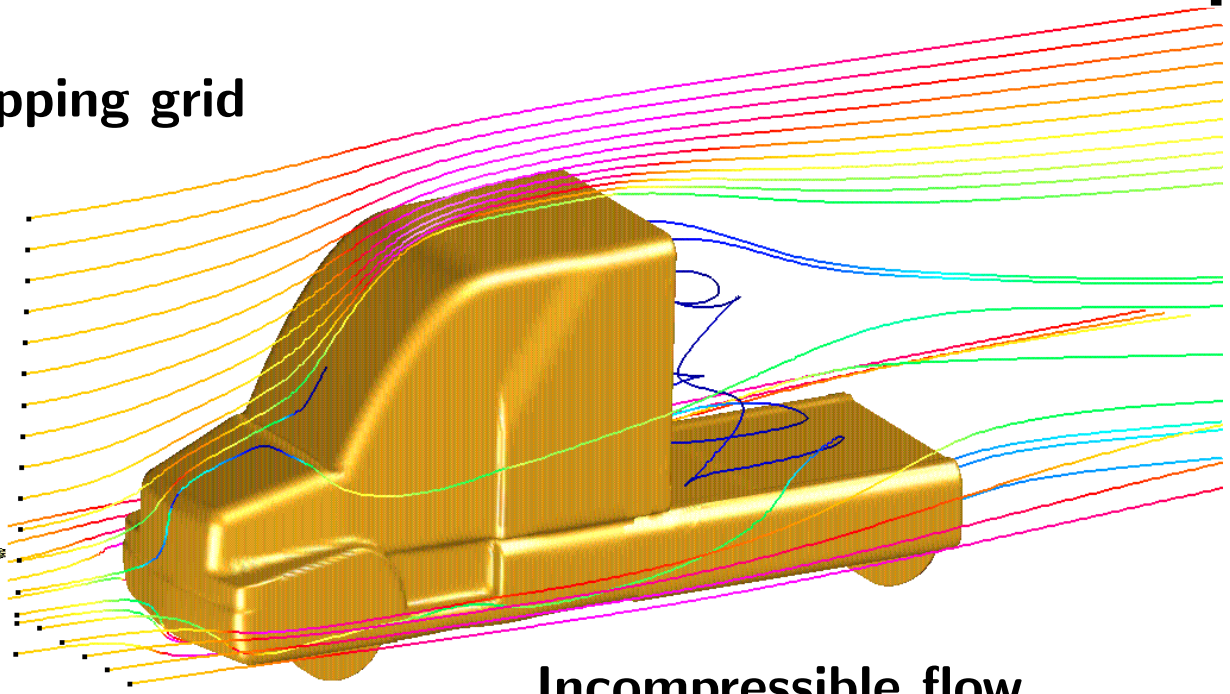
Cad geometry



Global triangulation



Overlapping grid



Incompressible flow.

Overture supports a high-level C++ interface (but is built mainly upon Fortran kernels):

Solve $u_t + au_x + bu_y = \nu(u_{xx} + u_{yy})$

```
CompositeGrid cg;  // create a composite grid
getFromADataBaseFile(cg,"myGrid.hdf");
floatCompositeGridFunction u(cg);  // create a grid function
u=1.;
CompositeGridOperators op(cg);  // operators
u.setOperators(op);
float t=0, dt=.005, a=1., b=1., nu=.1;
for( int step=0; step<100; step++ )
{
  u+=dt*( -a*u.x()-b*u.y()+nu*(u.xx()+u.yy()) ); // forward Euler
  t+=dt;
  u.interpolate();
  u.applyBoundaryCondition(0,dirichlet,allBoundaries,0.);
  u.finishBoundaryConditions();
}
```

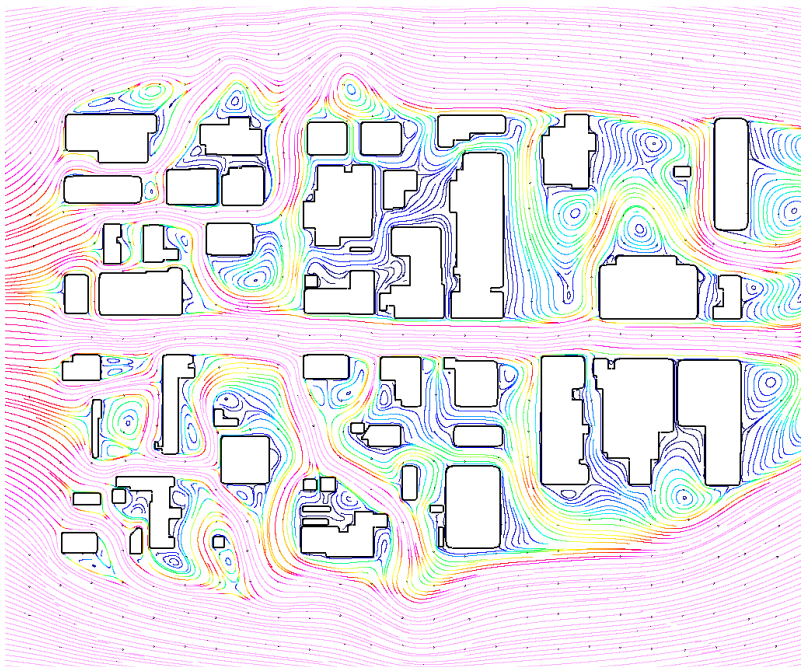
Current Projects with Overture

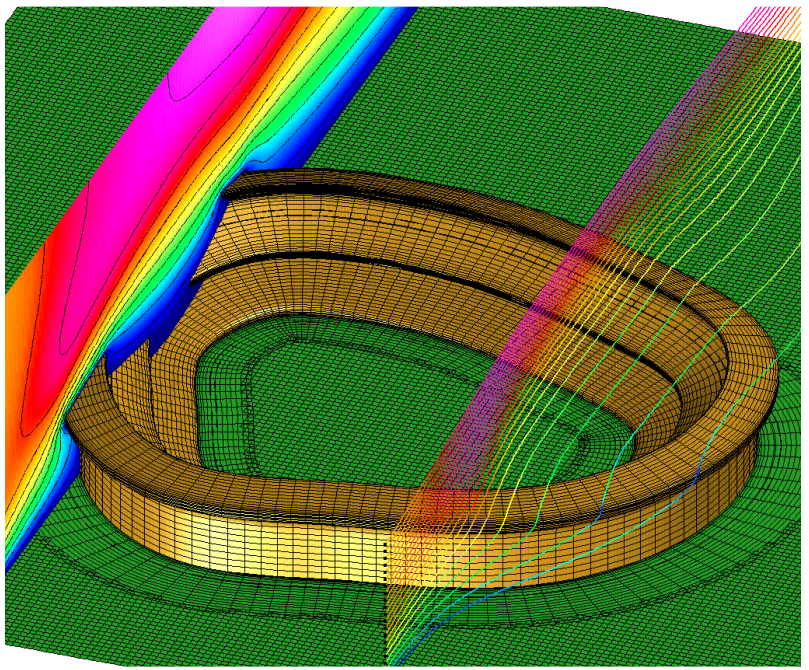
- Support for multi-physics problems, for example:
  - incompressible fluid flow coupled to solid heat transfer
  - compressible fluid flow coupled to solid mechanics
- Electromagnetics: time dependent Maxwell's equations.
- Parallel adaptive mesh refinement and parallel overlapping grid generation.
- Parallel multigrid algorithms for overlapping grids.
- High speed reactive flow with moving grids and AMR (Don Schwendeman (RPI))
- Compressible multiphase flows (Don Schwendeman (RPI))
- Compressible multi-material flows (Jeff Banks (SNL))
- Compressible axisymmetric flow with swirl (Kyle Chand)
- Compressible flow with ice formation (Graeme Leese, U. Cambridge).
- Einstein field equations (Philip Blakely, U. Cambridge)
- Flow on the surface of the eye (Kara Maki, U. Delaware).

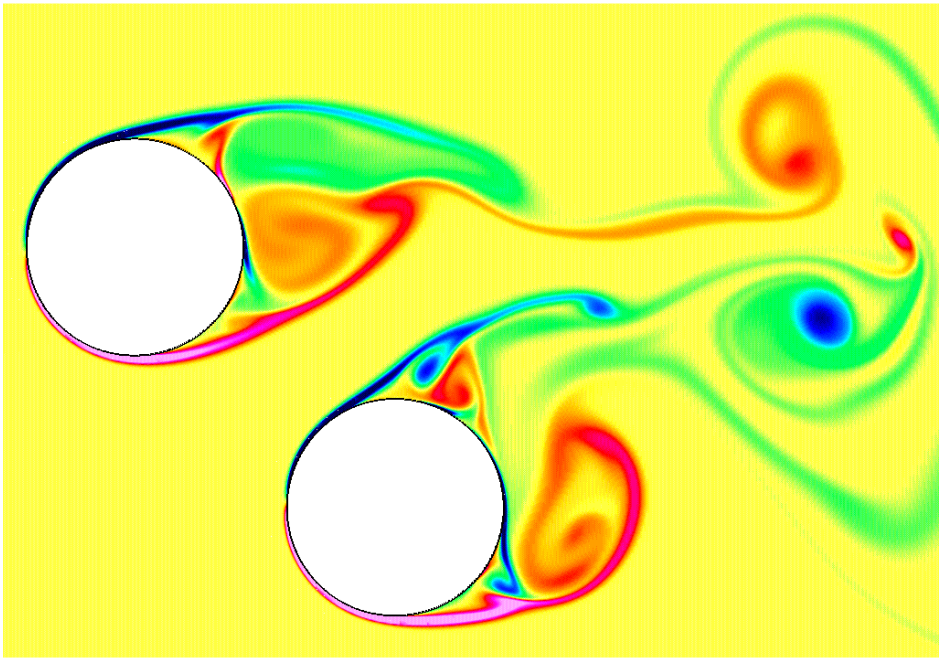
Incompressible Flow and Rigid-body Motion

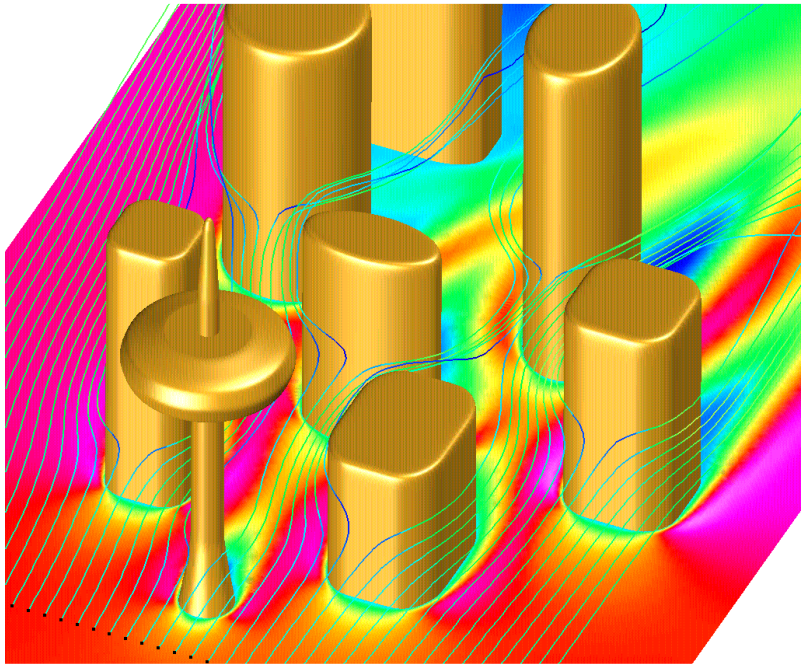
Reference:

WDH., *A Fourth-Order Accurate Method for the Incompressible Navier-Stokes Equations on Overlapping Grids*, J. Comput. Phys, **113**, no. 1, (1994) 13–25.





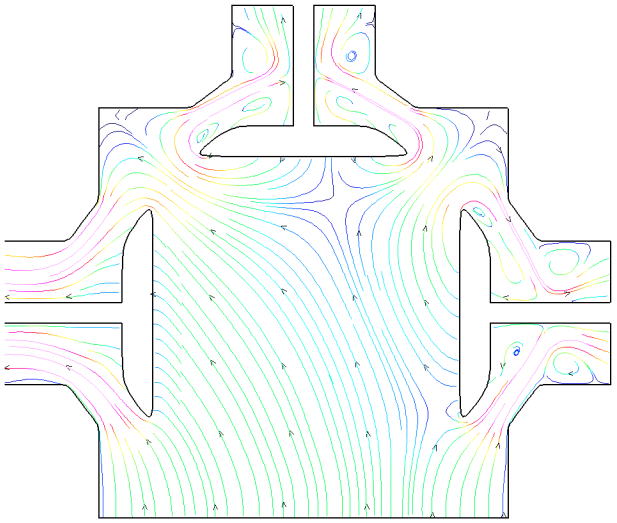




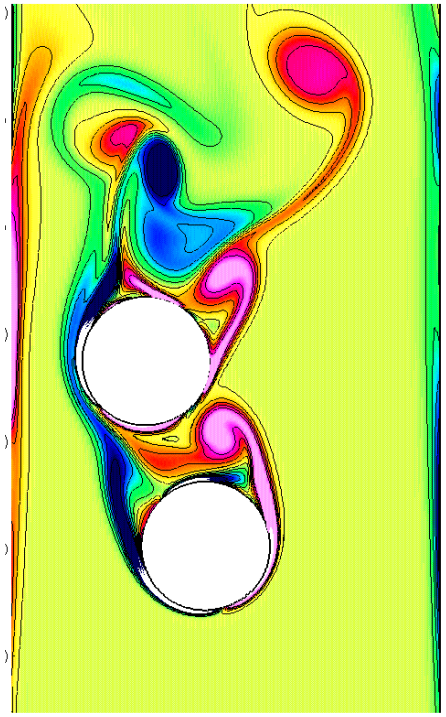
Incompressible flow computations with cgins.

Moving Overlapping Grids

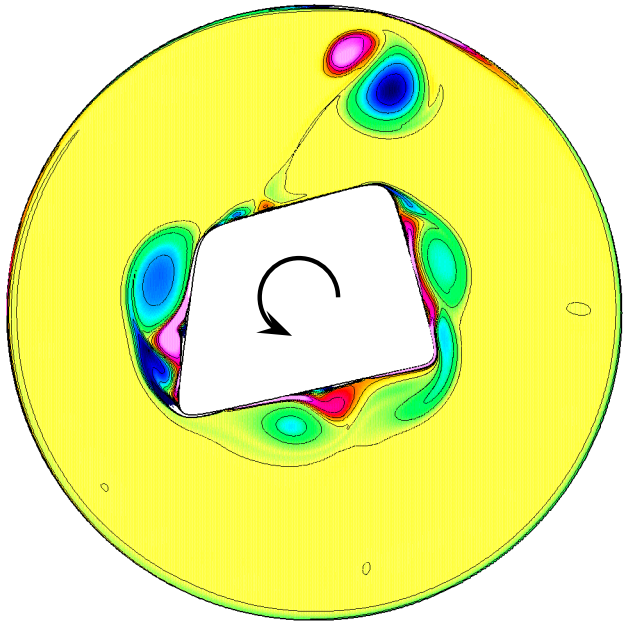
- ♢ Boundary fitted component grids are used to discretize each moving body.
- ♢ Grids move at each time step according to some governing equations.
- ♢ Overlapping connectivity information is updated by Ogen (interpolation points, discretization points, unused points).
- ♢ Solution values at **exposed points** are interpolated at previous time levels.
- ♢ Issue: Detection and treatment of collisions – elastic/in-elastic collisions
- ♢ Issue: Bodies that get very close – how should the grids interpolate



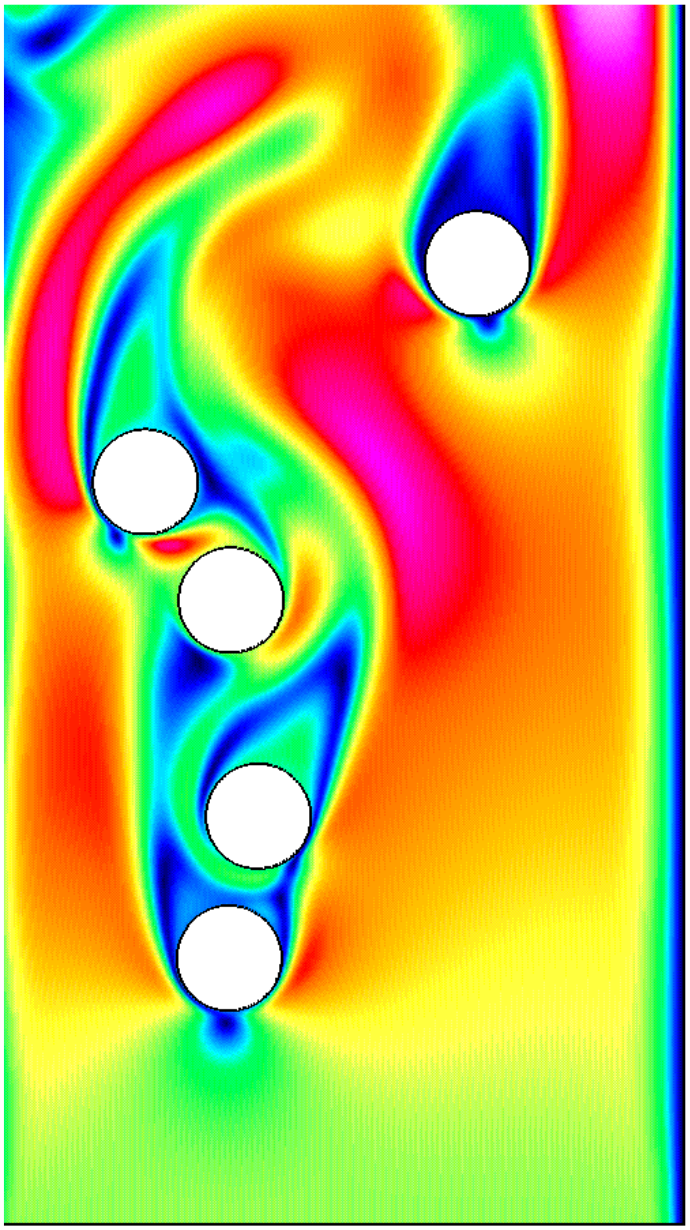
Moving valves (INS)

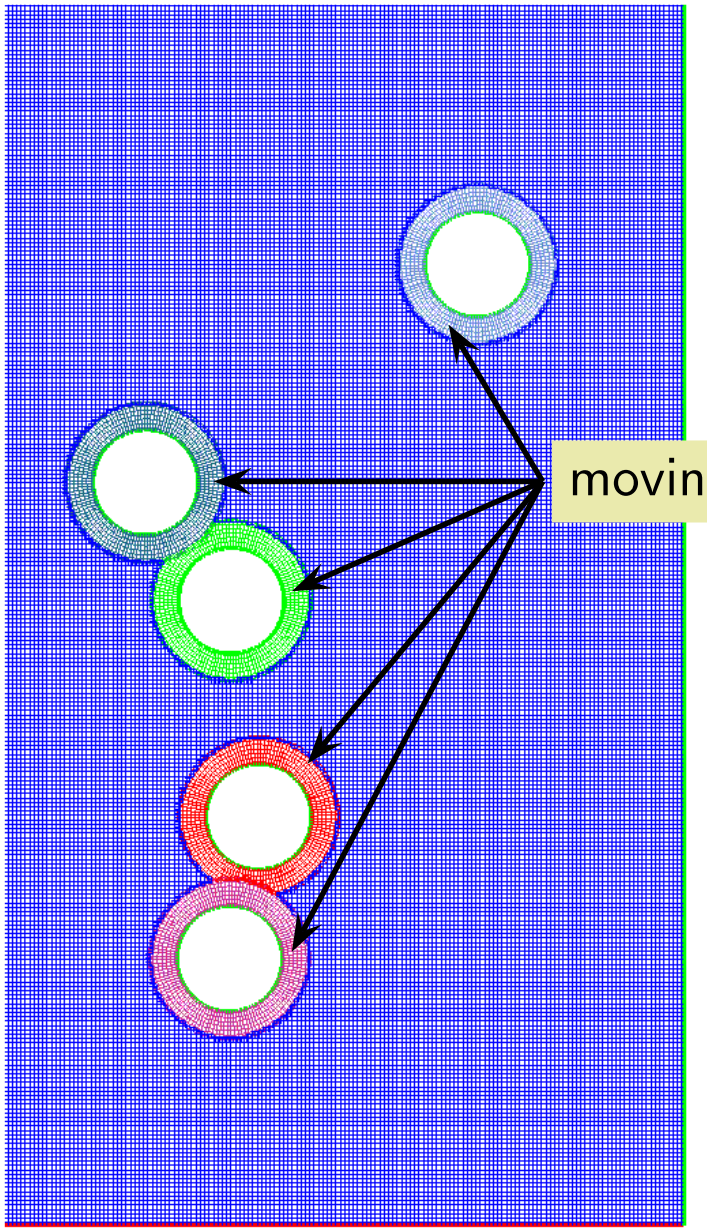


Falling cylinders (INS)



Rotating body (INS)





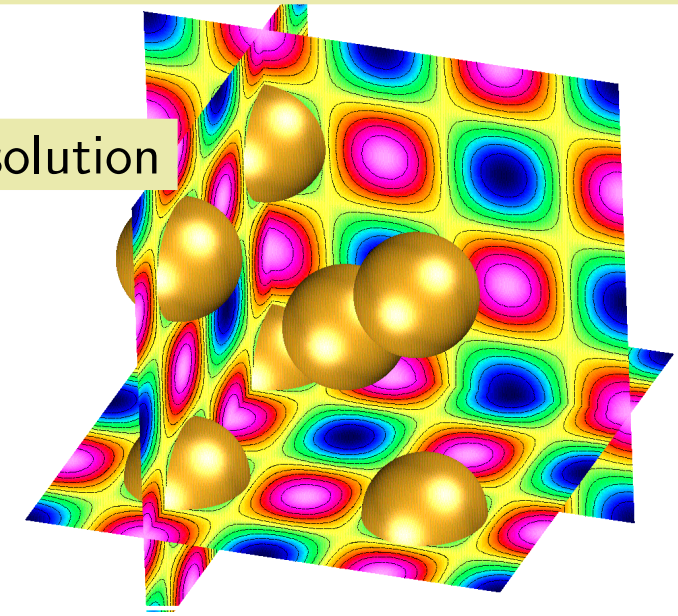
Falling cylinders in an incompressible flow

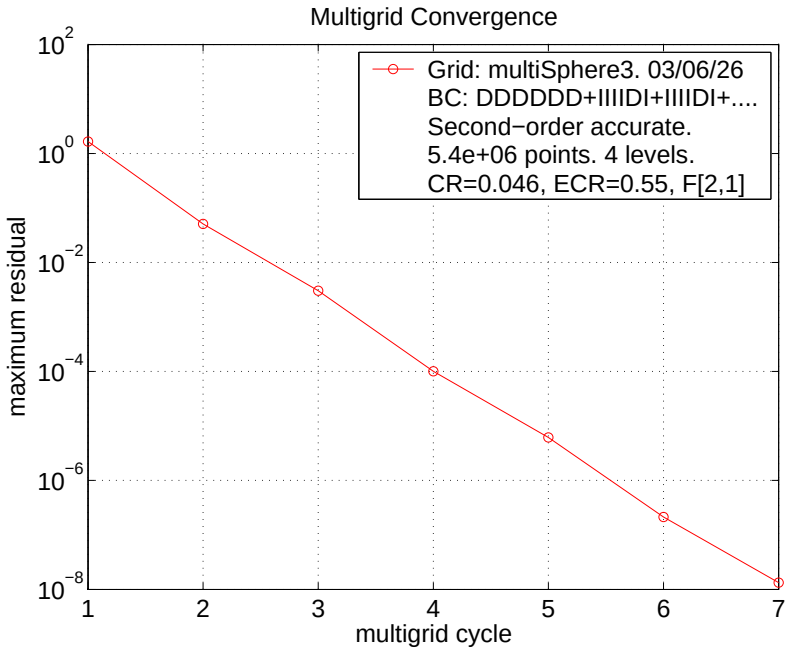
Multigrid for Overlapping Grids

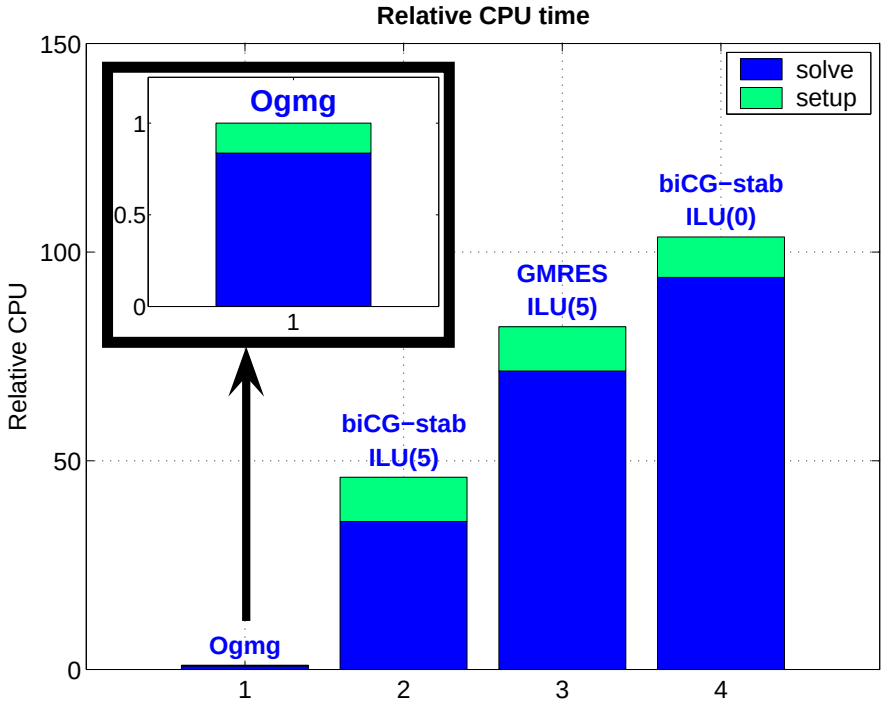
Reference:

WDH., *On Multigrid For Overlapping Grids*, SIAM J. Sci. Comput. **26**, no. 5, (2005) 1547–1572.

Multigrid solution to Poisson's equation, 5.4 million grid points







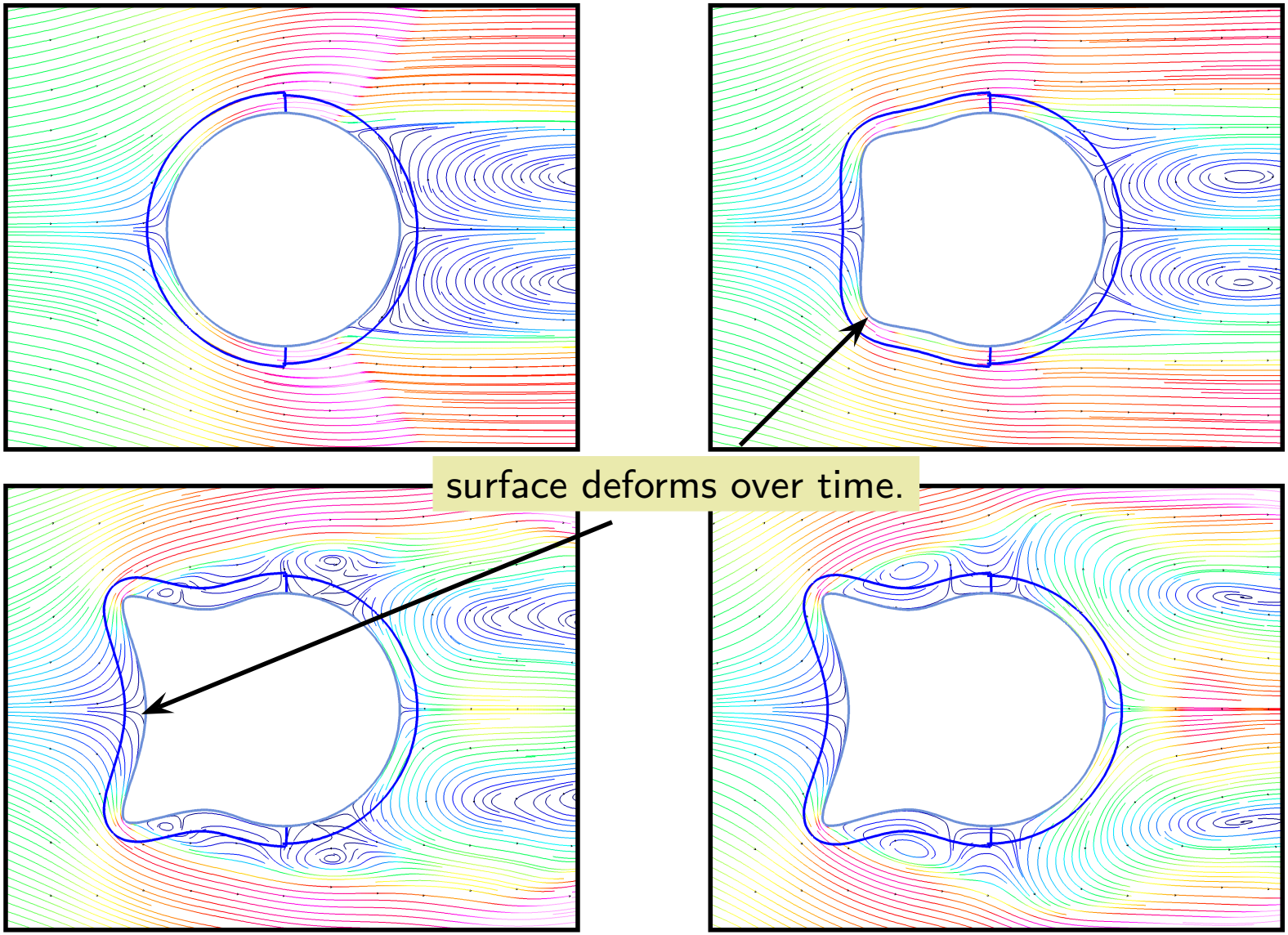
Mesh independent convergence rates

New adaptive MG for overlapping grids.

In comparison to Krylov solvers multigrid is an order of magnitude faster and uses an order of magnitude less storage

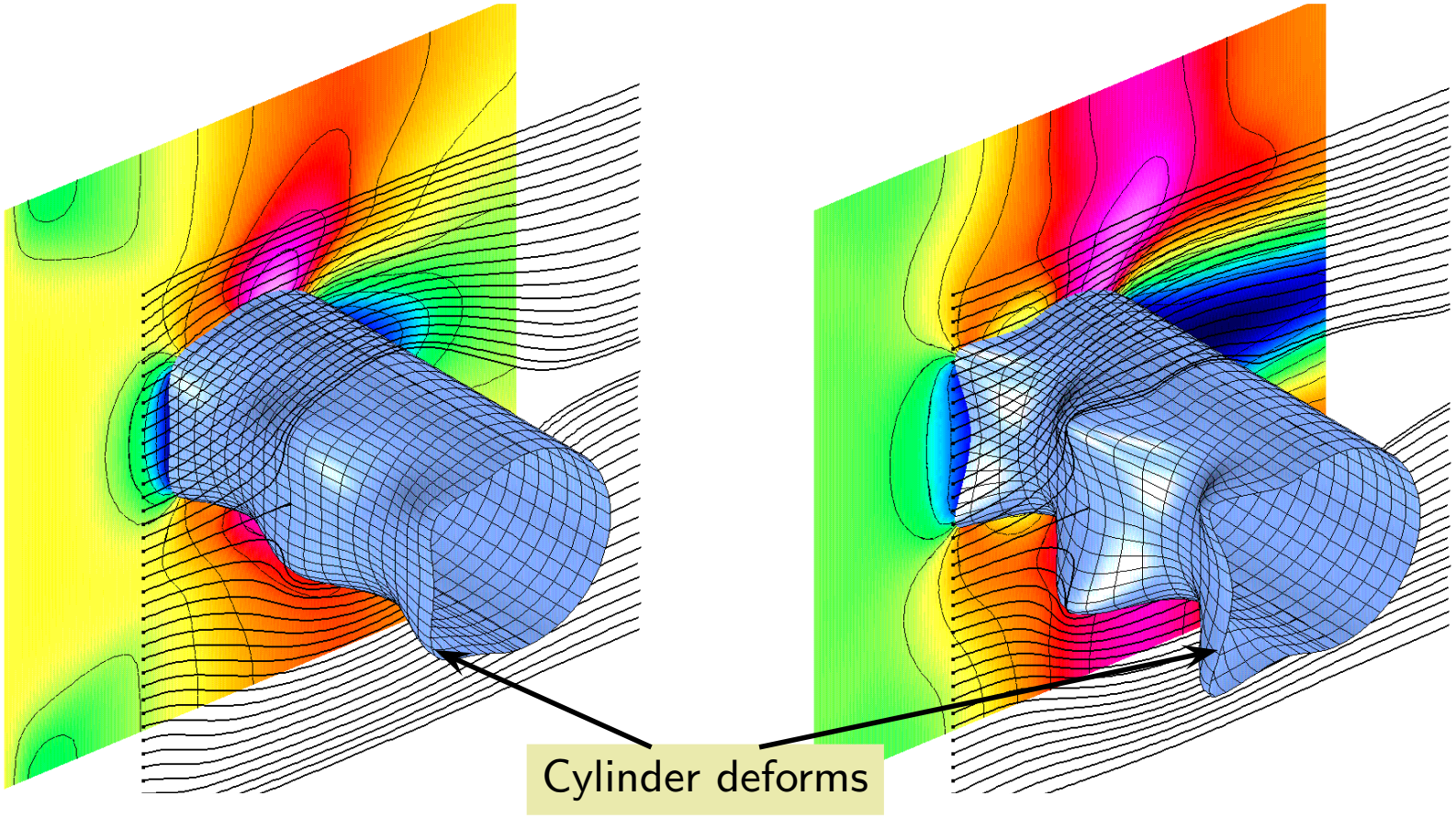
Deforming Grids

Modeling Deforming Geometry with Overlapping Grids



Streamlines of a compressible flow around a deforming boundary.

Deforming Body Applications



Compressible flow past a deforming cylinder. The surface of the cylinder deforms over time to mimic the growth of ice. Only a subset of the grid lines are shown.

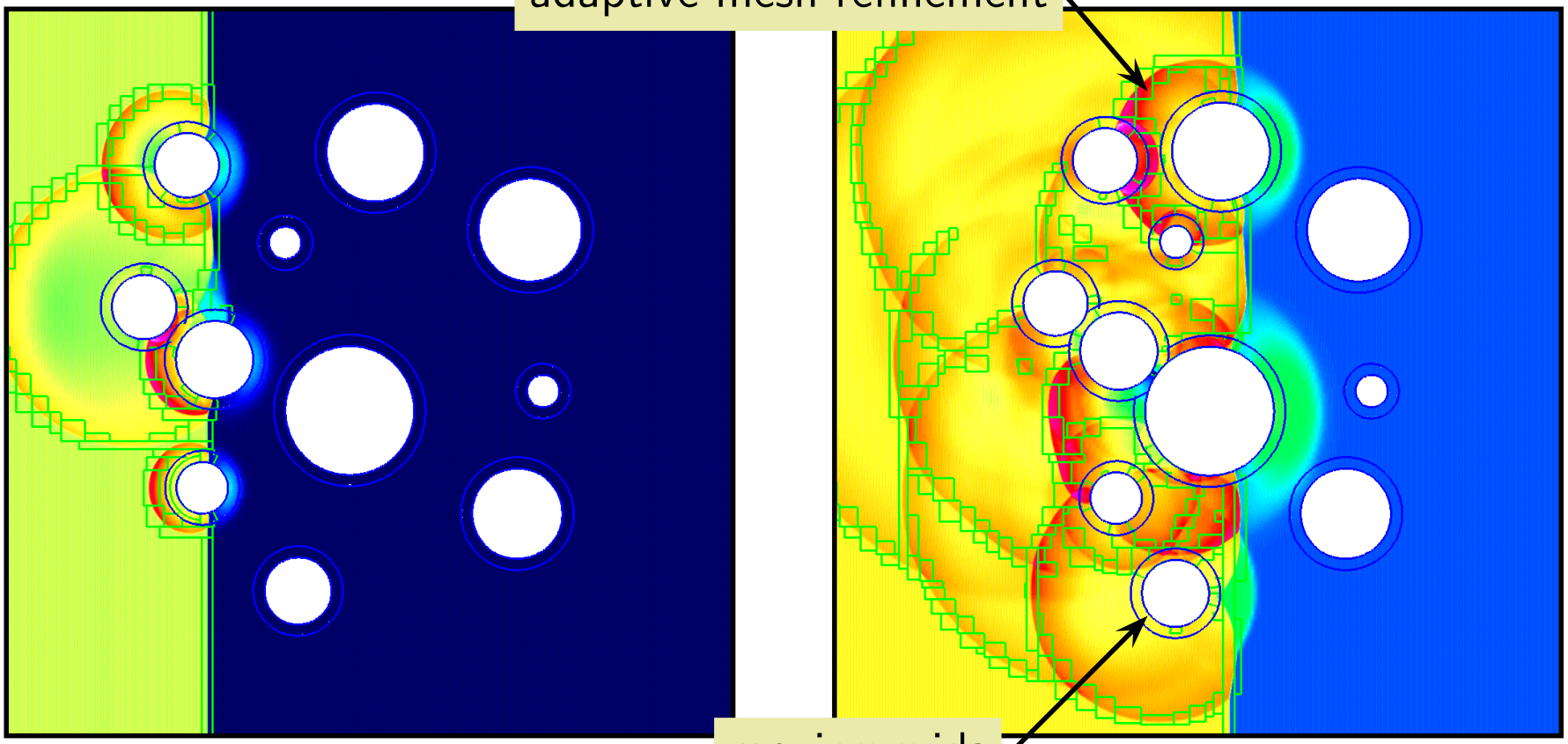
High-Speed Reactive Flow with Rigid-body Motion

References:

WDH., D. W. Schwendeman, *Moving Overlapping Grids with Adaptive Mesh Refinement for High-Speed Reactive and Nonreactive Flow*, J. Comp. Phys. **216** (2005) 744-779.

WDH., D. W. Schwendeman, *An adaptive numerical scheme for high-speed reactive flow on overlapping grids*, J. Comp. Phys. **191** (2003) 420–447.

Moving geometry and AMR



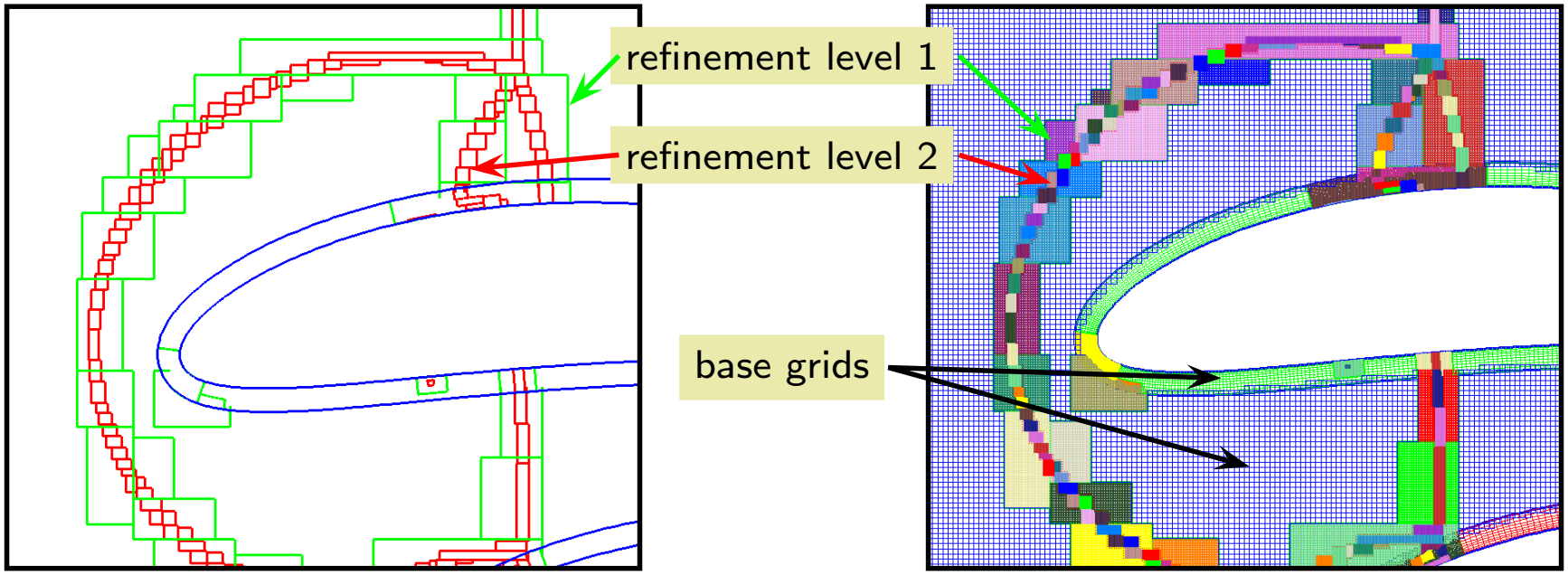
A shock hitting a collection of cylinders (density).

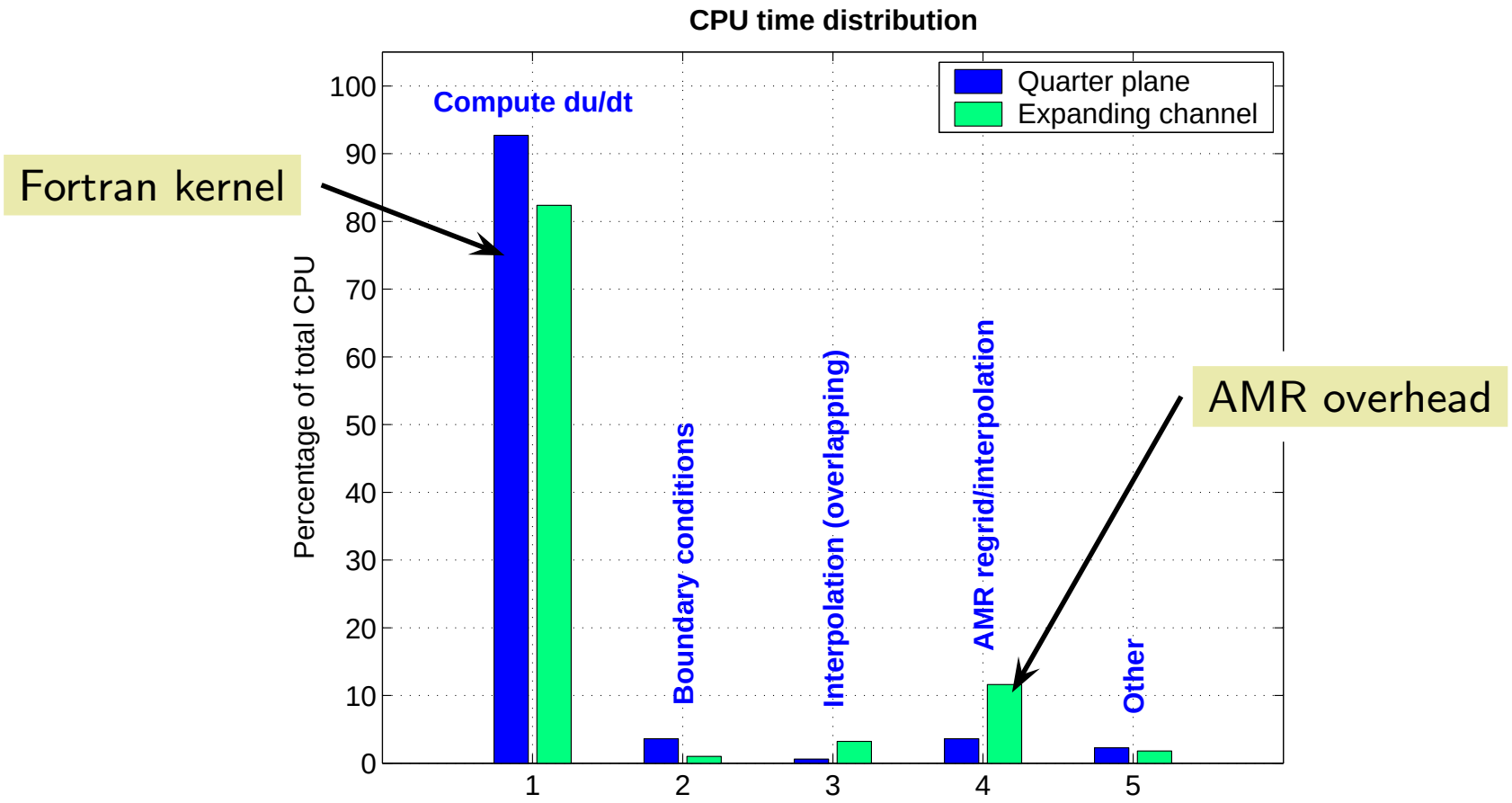
Summary of the Solution Method

- **Euler-equations:** solved with a second-order accurate Godunov Solver.
- **Chemistry:** solved with a Runge-Kutta error-control scheme (Strang Splitting)
- **Newton Euler Equations (rigid body motion)**: solved with a second-order accurate predictor-corrector scheme (leap-frog predictor, trapezoidal-rule corrector).

Block Structured Adaptive Mesh Refinement and Overlapping Grids

- Refinement patches are generated in the parameter space of each component grid (base grid).
- Refinement patches are organized in a hierarchy of *refinement levels*.
- Error estimators determine where refinement is needed.
- AMR grid generation (Berger-Rigoutsos algorithm) builds refinement patches based on the error estimate.
- refinement grids may interpolate from refinement grids of different base grids.
- The key issue is efficiency.

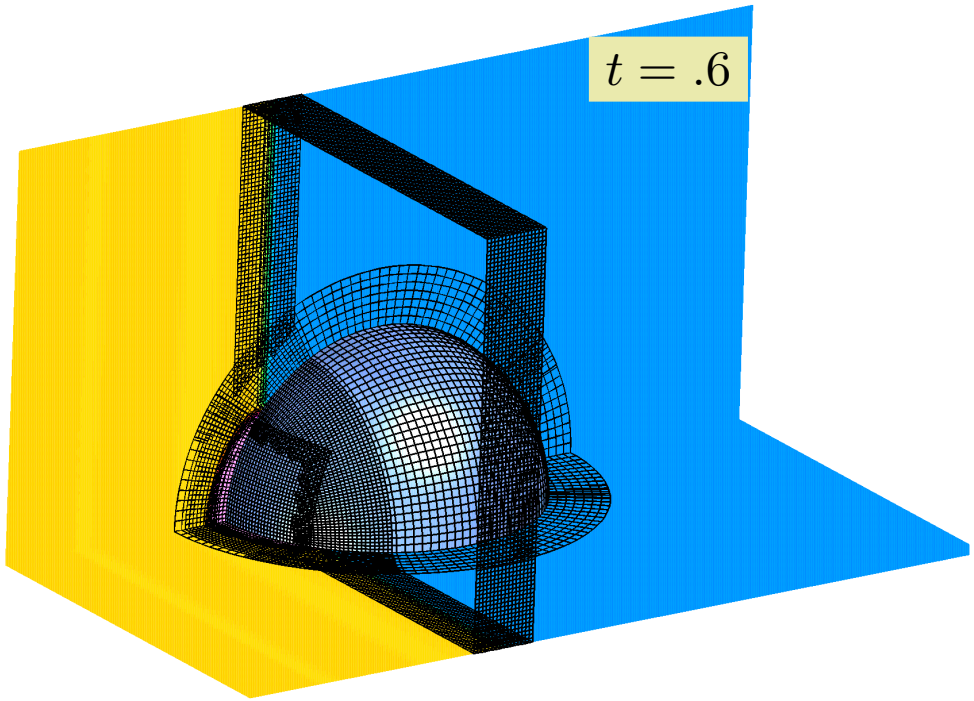


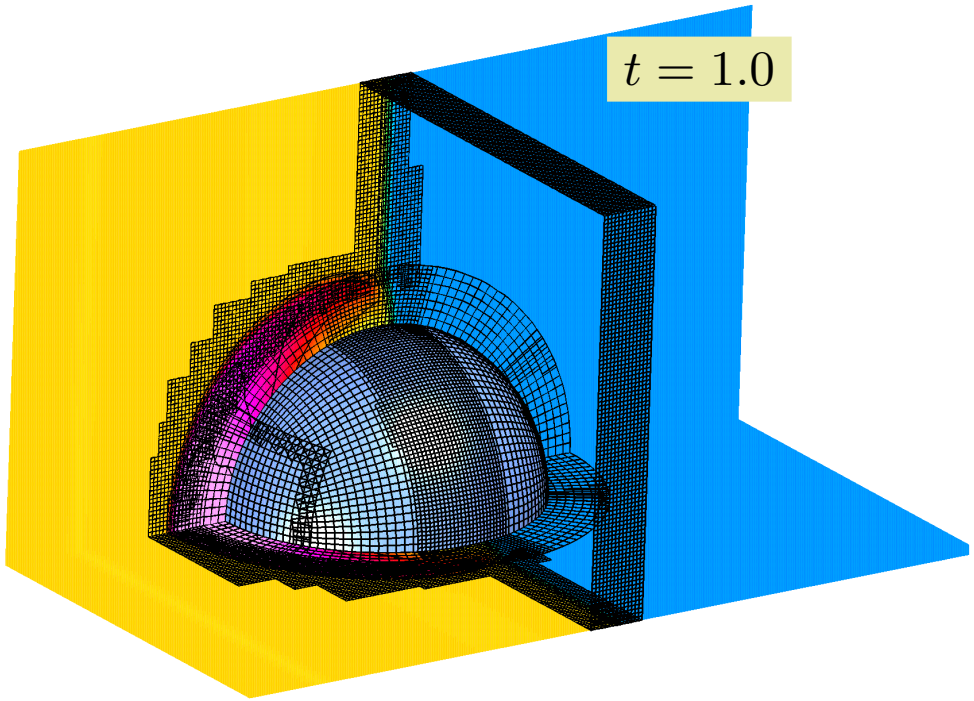


	Quarter plane	Expanding channel
time steps	12418	21030
seconds per step	14.94	13.96
grids (min,ave,max)	(2, 57, 353)	(5, 274, 588)
points (min,ave,max)	(2.0e5, 9.2e5, 1.9e6)	(1.2e5, 6.4e5, 1.3e6)

Overlapping grid AMR performance on two detonation problems.

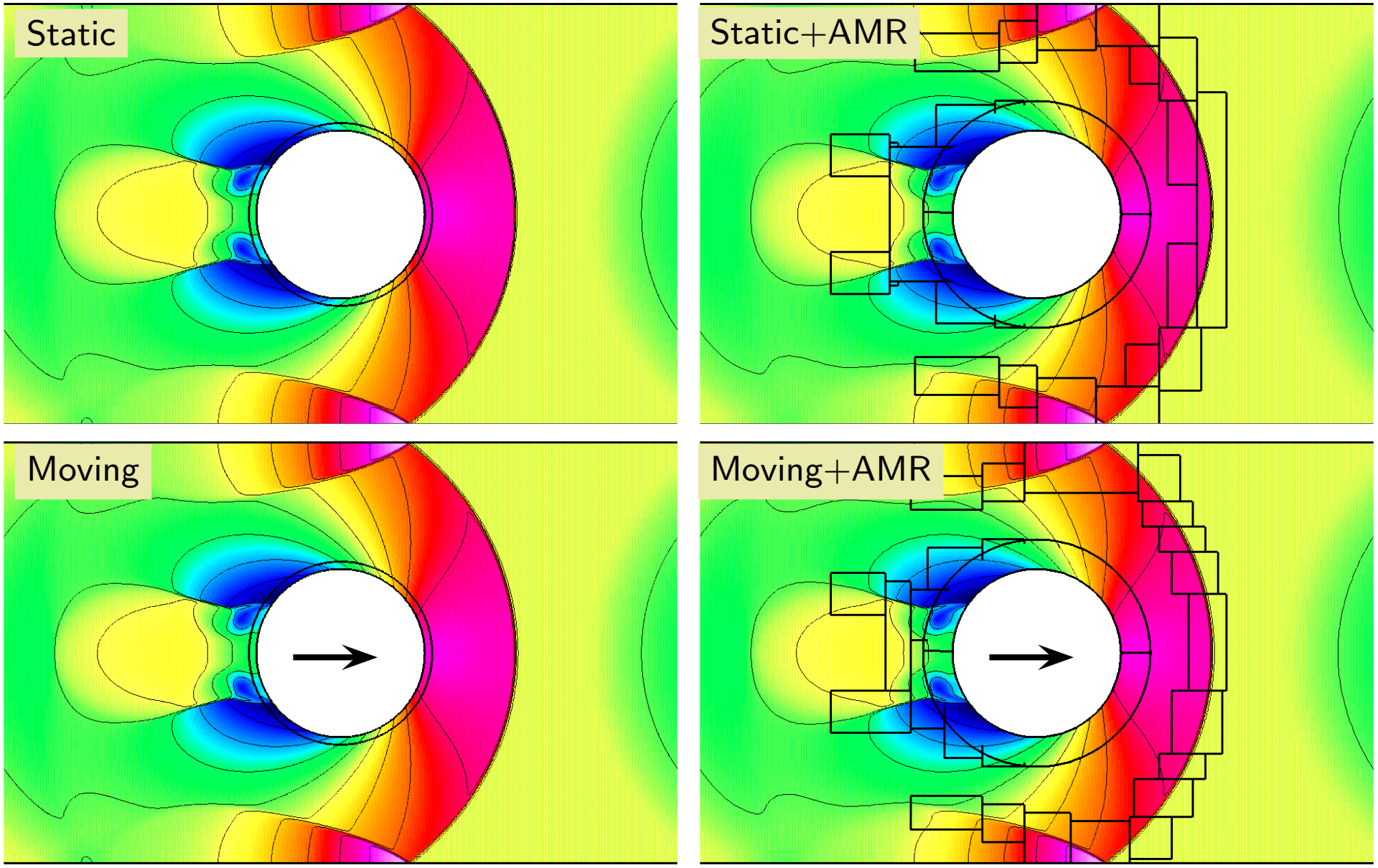
Parallel Adaptive Mesh Refinement on Overlapping Grids

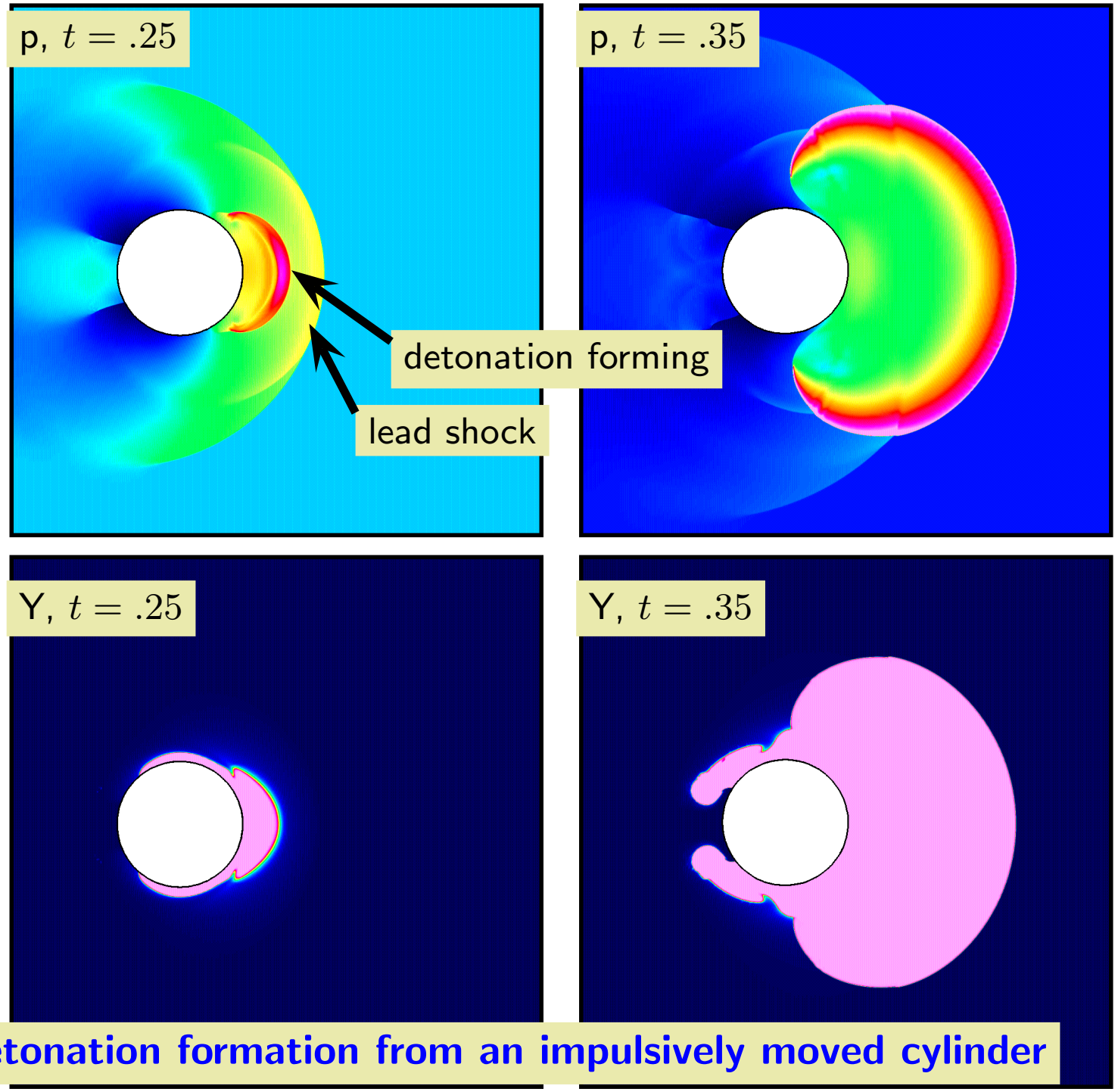




A shock hitting a sphere.

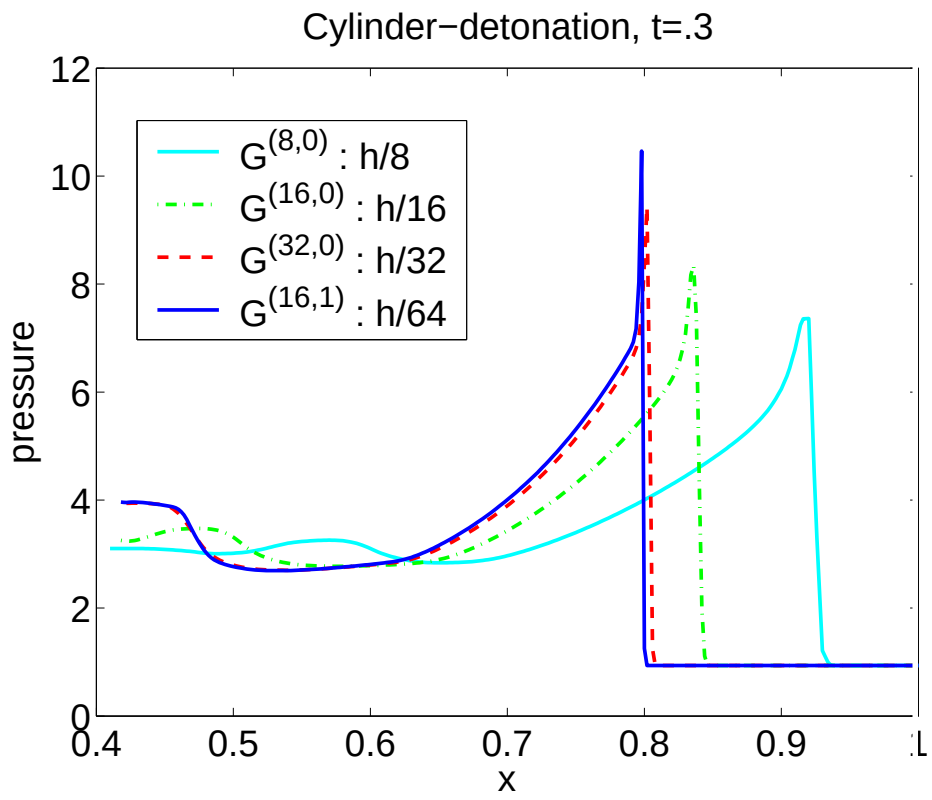
Comparing different approaches to solving the same problem

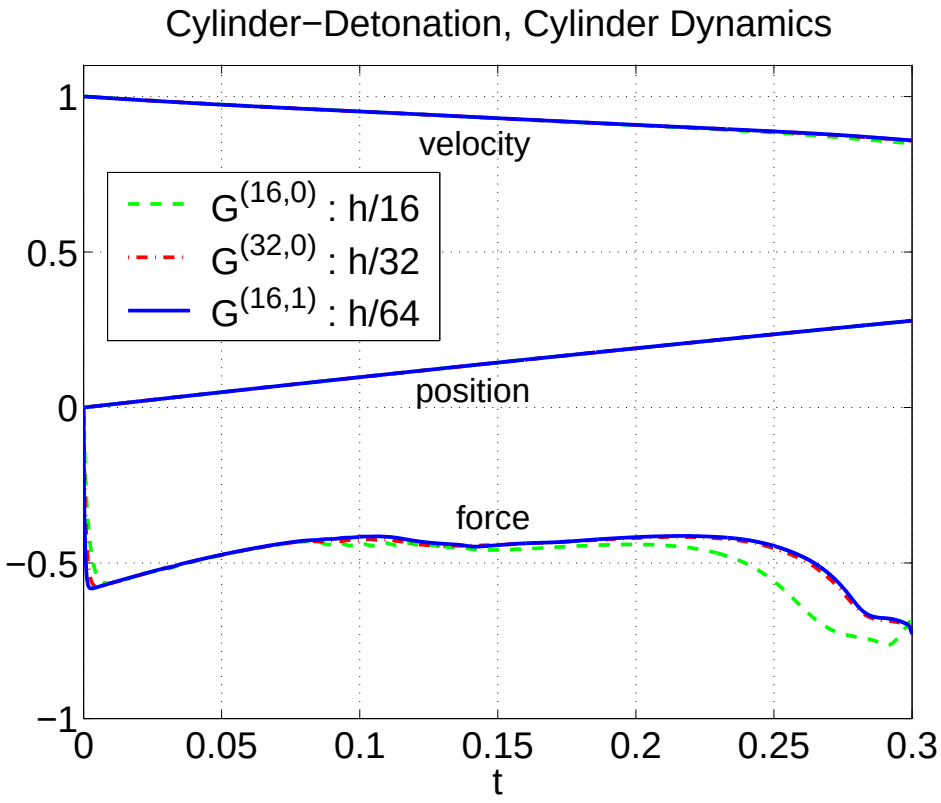




Detonation formation from an impulsively moved cylinder

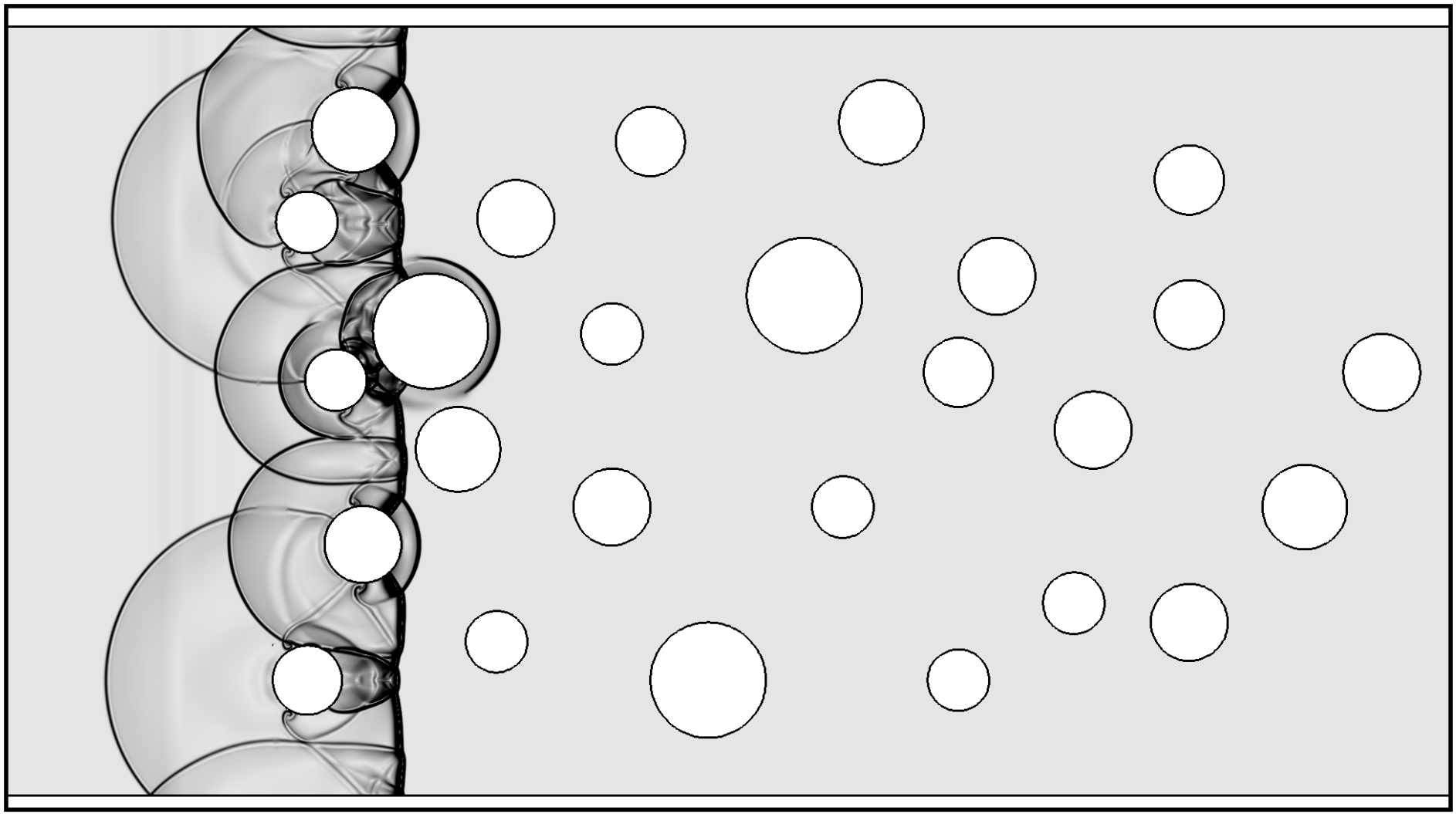
Detonation formation from an impulsively moved cylinder
Grid Convergence Studies



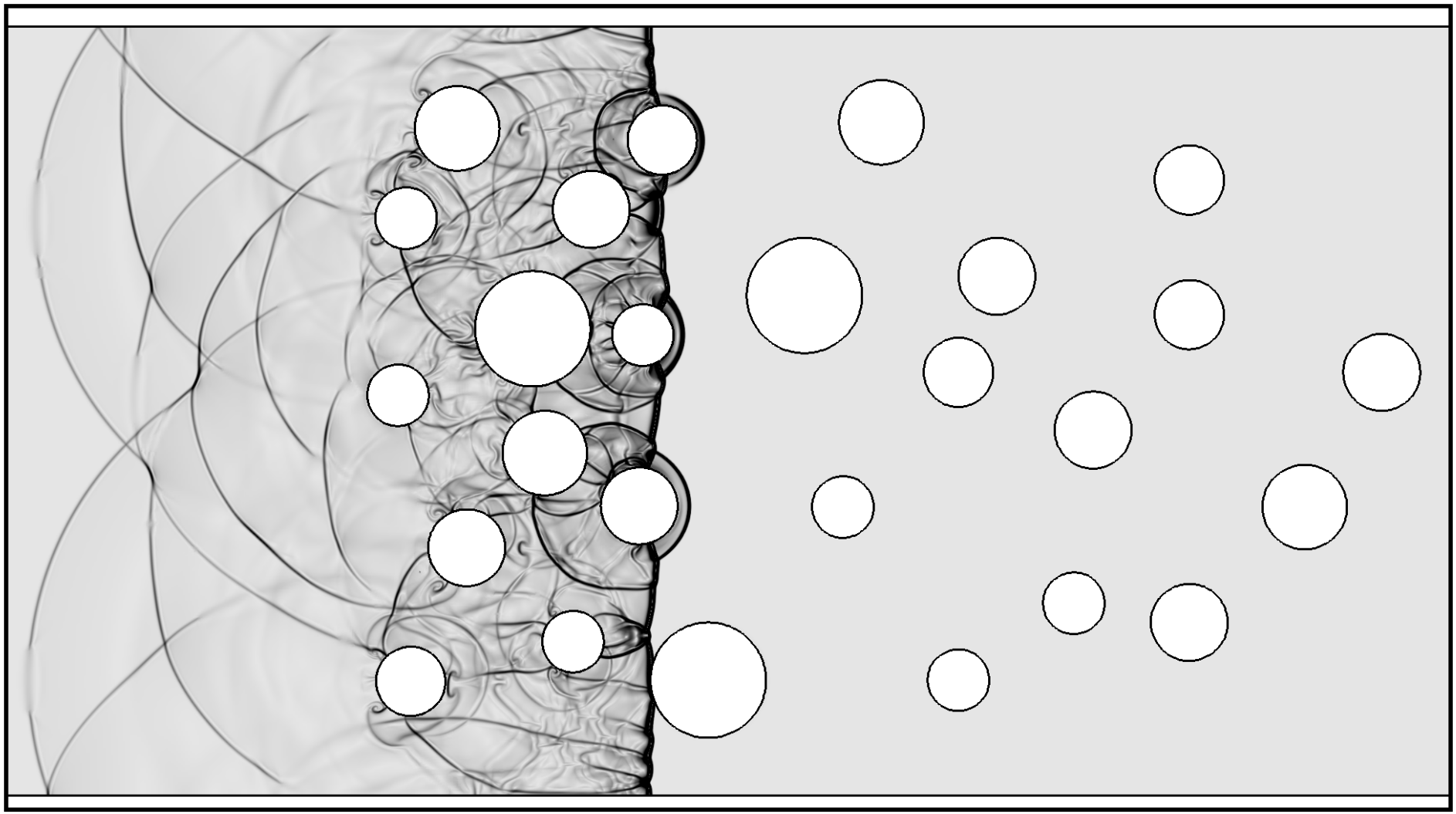


Pressure ahead of the cylinder ($y = 0$)

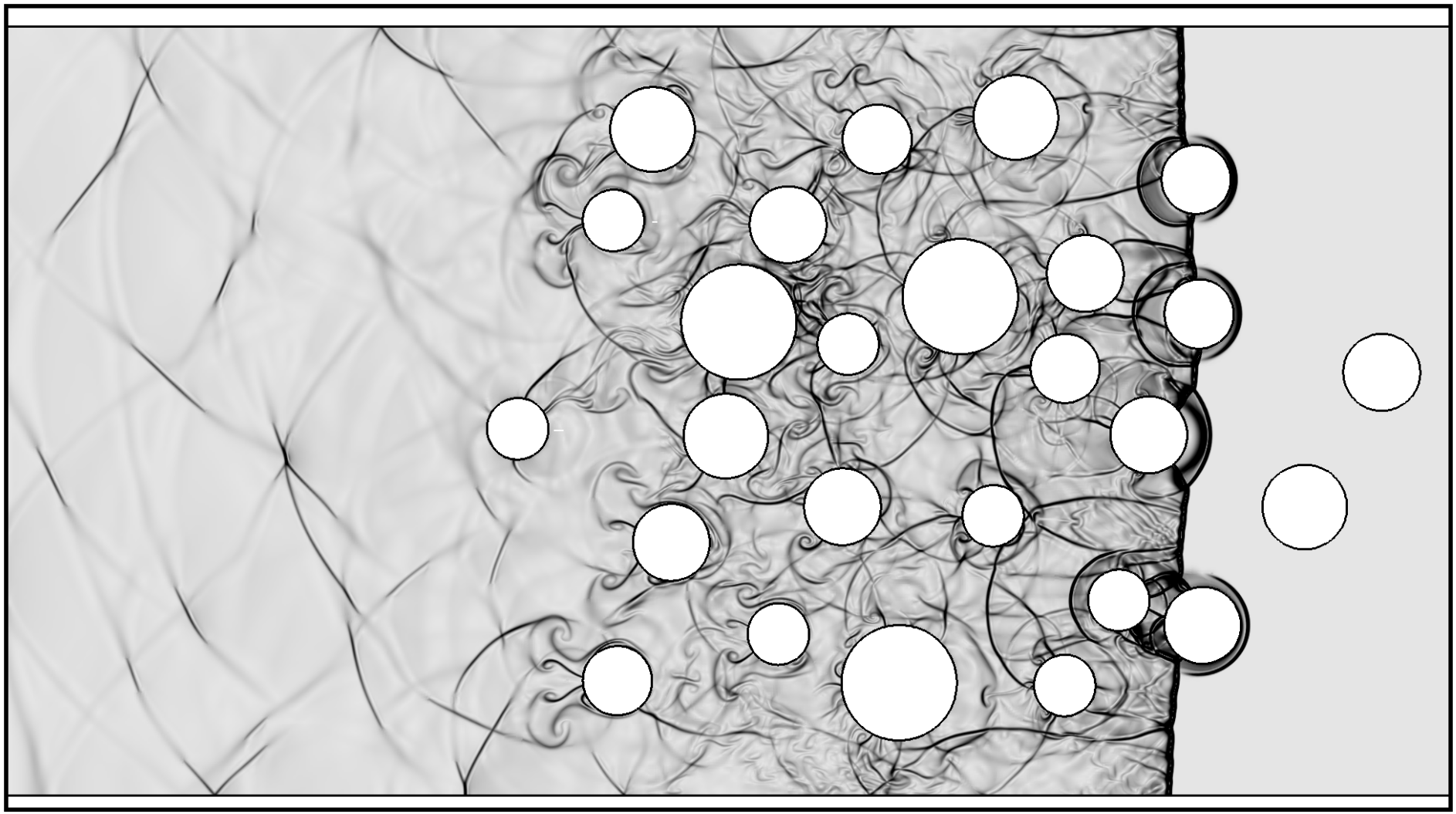
Detonation hitting a collection of cylinders, $t = .3$

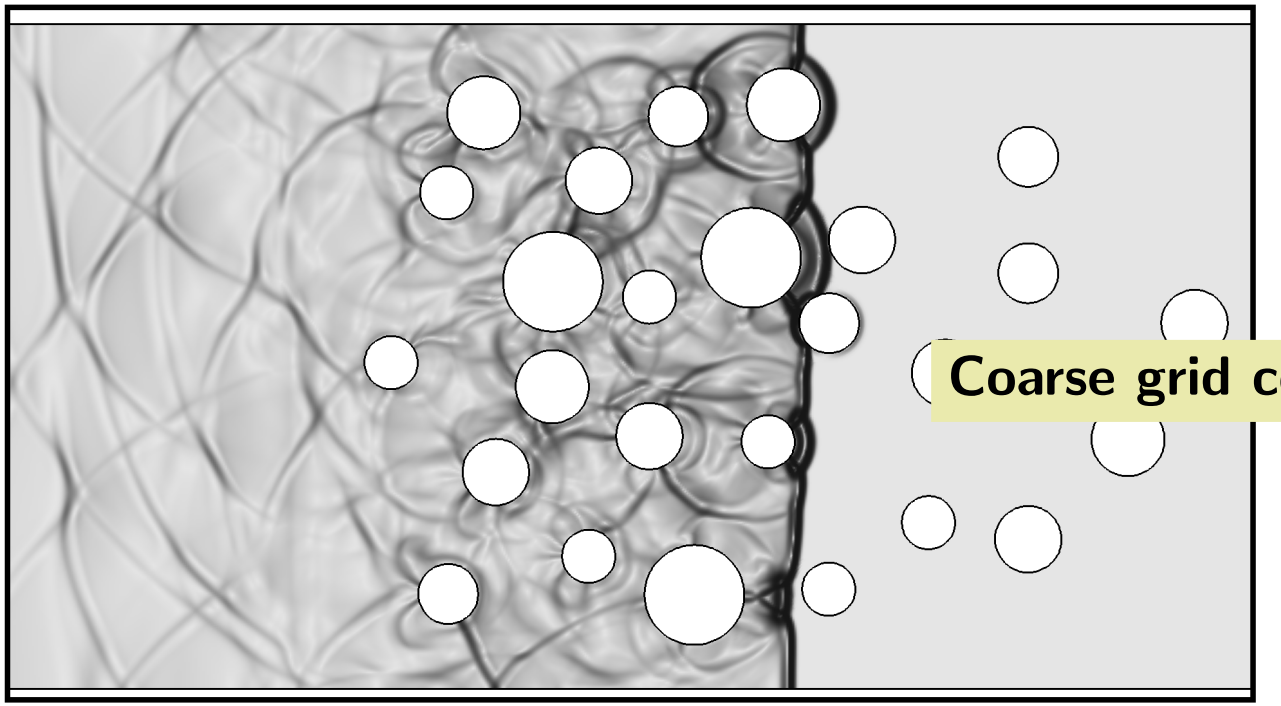


Detonation hitting a collection of cylinders, $t = .5$

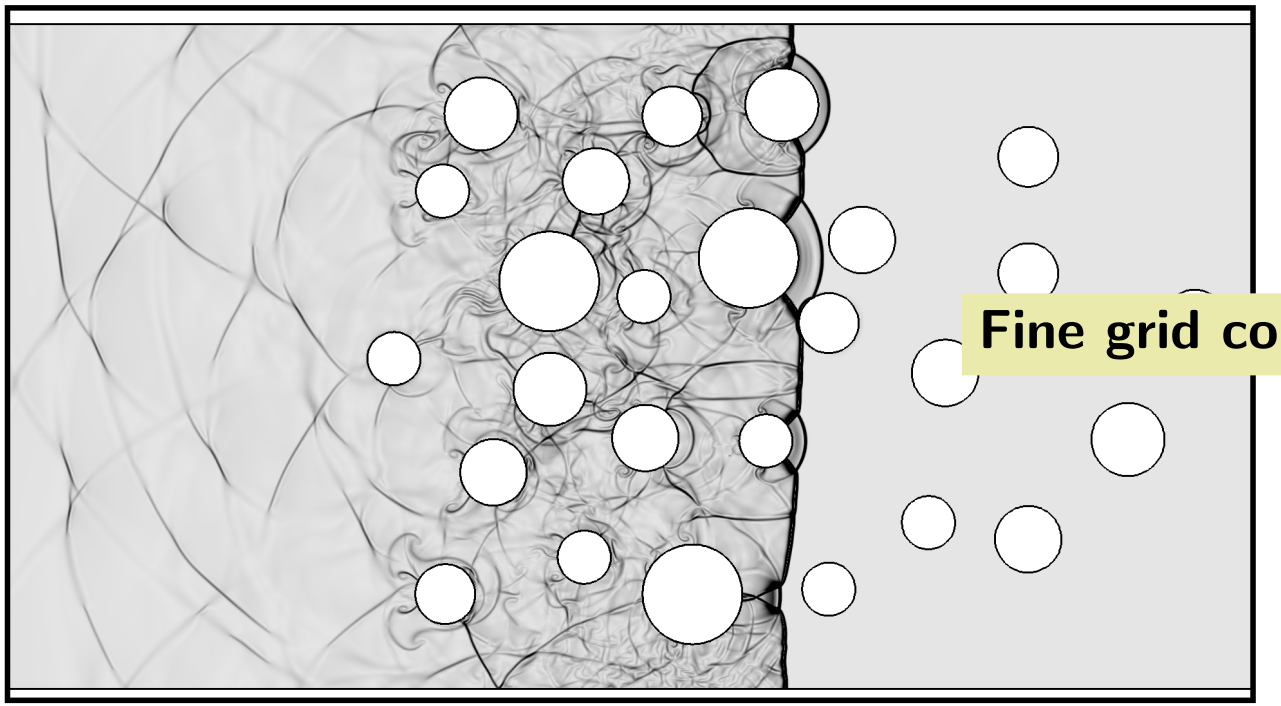


Detonation hitting a collection of cylinders, $t = .9$





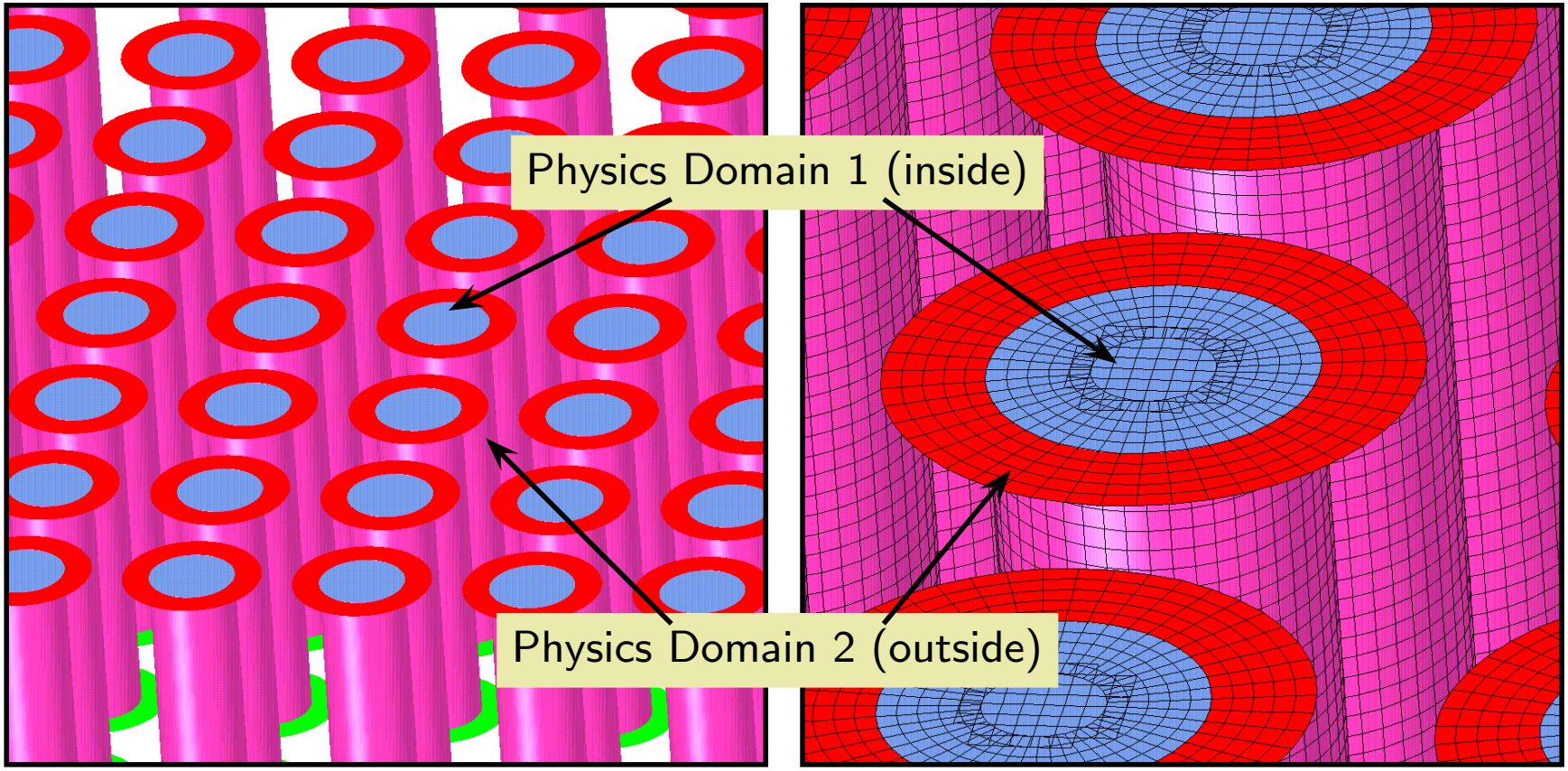
Coarse grid computation.



Fine grid computation.

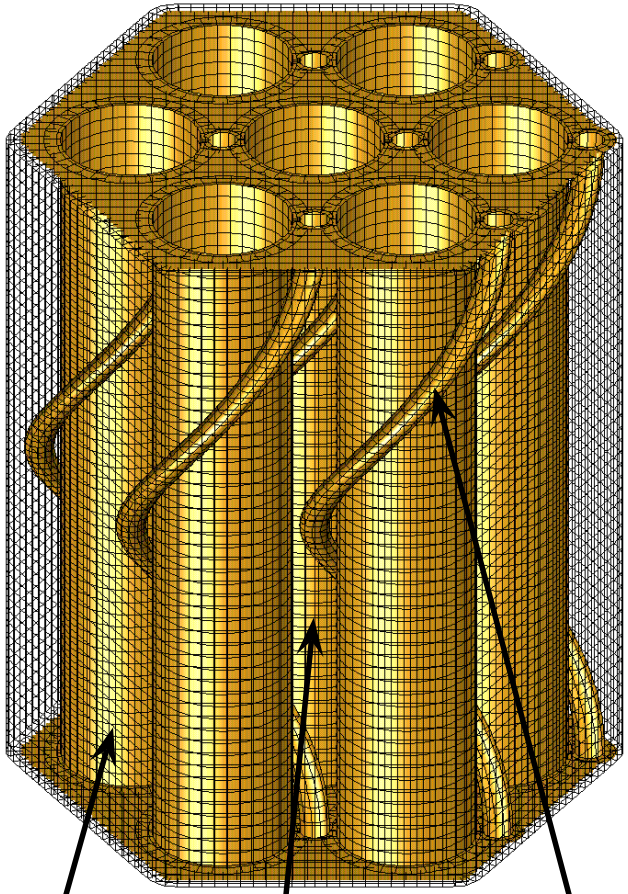
Multi-domain and Multi-Physics Applications

Multi-Physics and Multi-Material Applications



An overlapping grid for a lattice of cylinders modeling a photonic bad gap device (Maxwell's equations) or a heat exchanger (fluid-flow solid-heat transfer).

Subassembly Grids for Pins with Wire Wraps

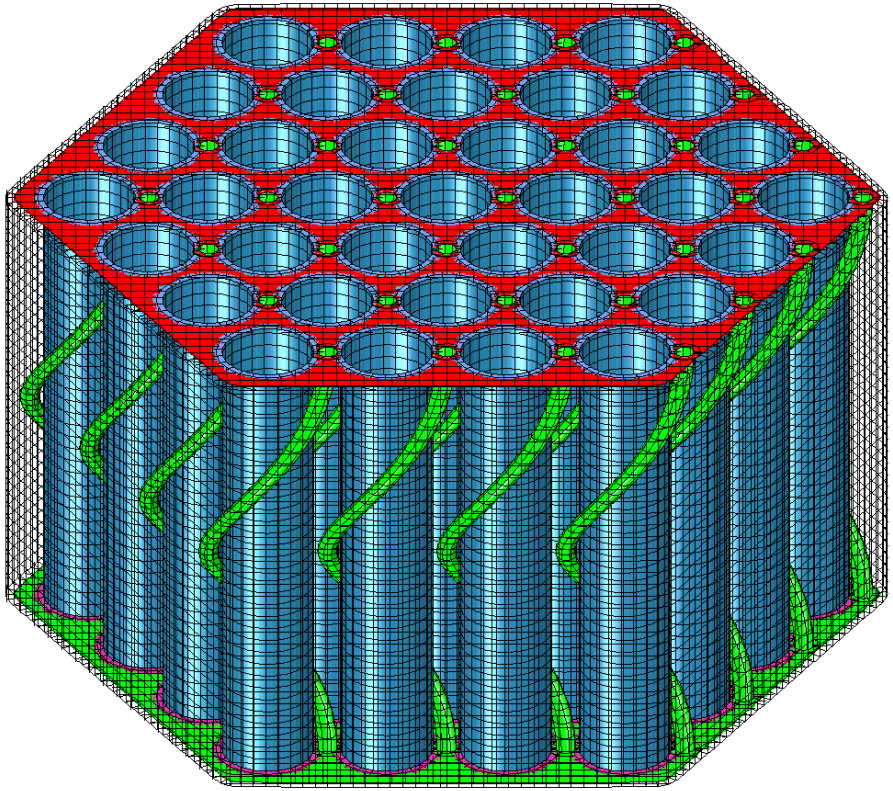


fuel pin

fluid channel

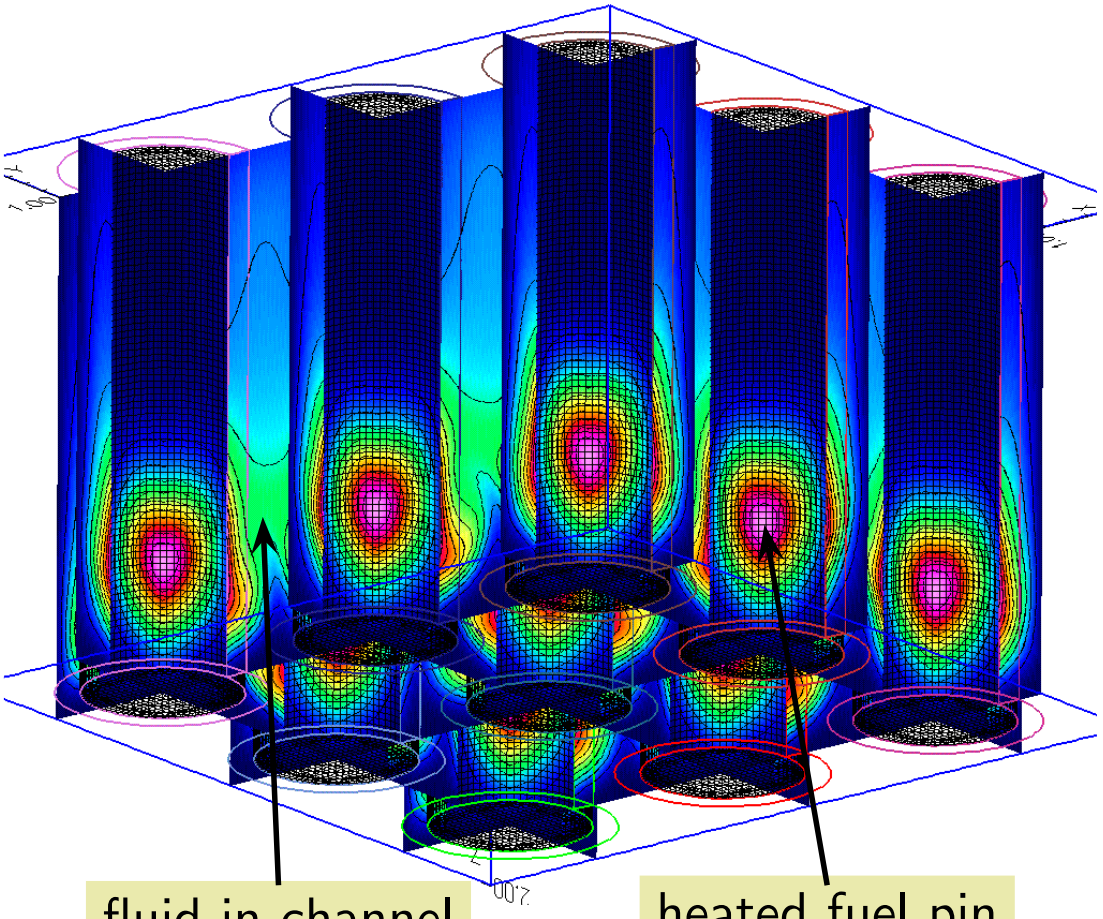
wire wrap

overlapping grid



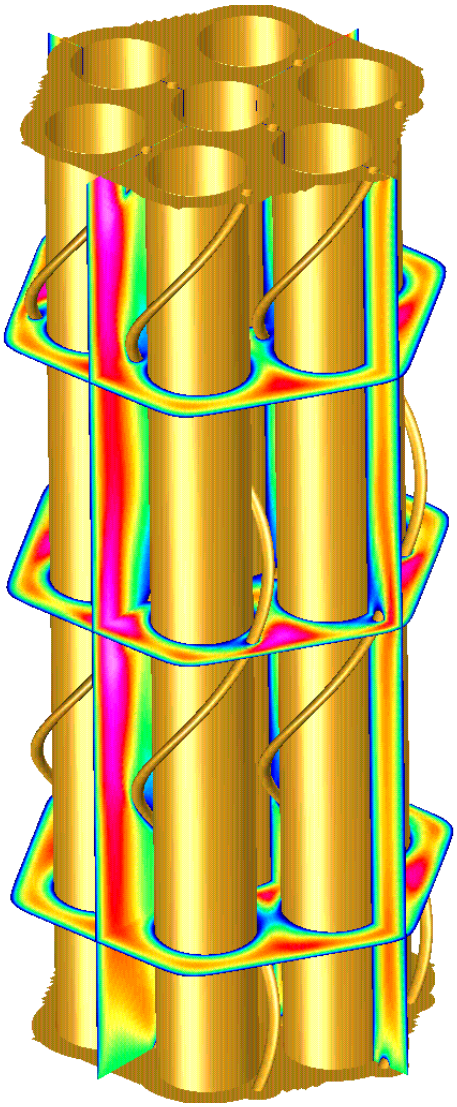
37 Pin subassembly

Subassembly Thermal-Hydraulics Flow



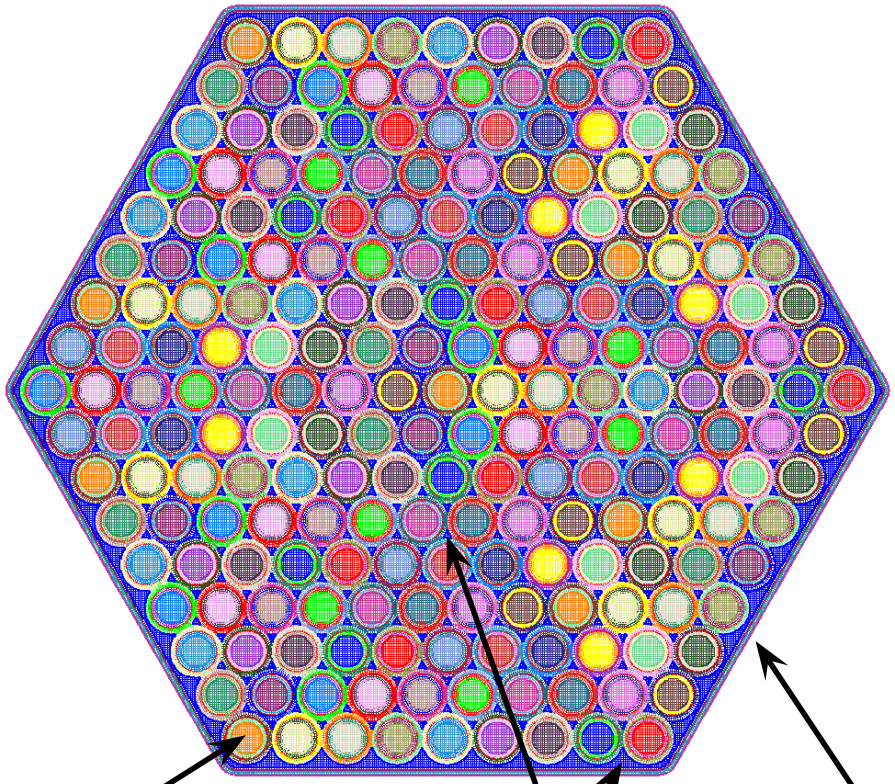
fluid-solid conjugate heat transfer

flow



T-H flow, 7 pins with wires

Cross-Section of a 217 Pin Fuel-Subassembly

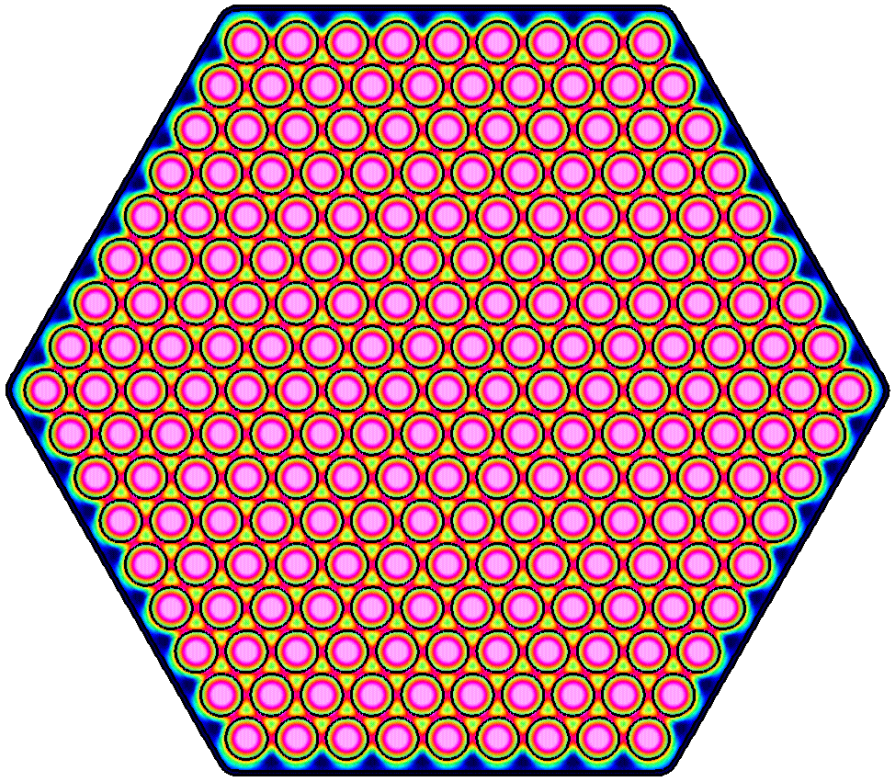


solid fuel pin

fluid channel

solid duct

overlapping grid for fluid and solid domains



conjugate heat transfer (2D)

The model for distributed parallel computing in Overture

- ♢ Grids can be distributed across one or more processors.
- ♢ Distributed parallel arrays using P++ (K. Brislawn, B. Miller, D. Quinlan)
- ♢ P++ uses Multiblock PARTI (A. Sussman, G. Agrawal, J. Saltz) for block structured communication with MPI (ghost boundary updates, copies between different distributed arrays)
- ♢ A special parallel overlapping grid interpolation routine is used for overlapping grid interpolation.

P++ : parallel multi-dimensional arrays

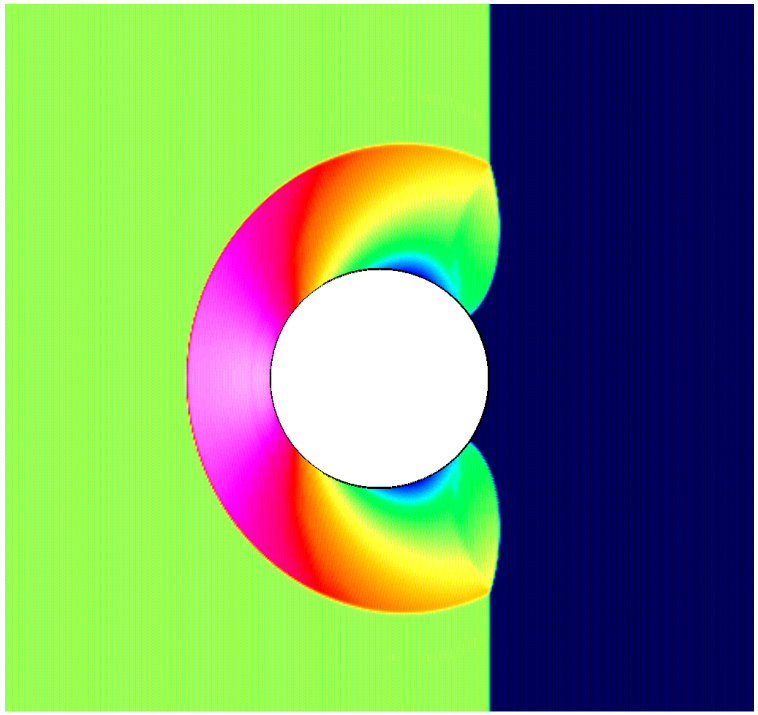
```
Partitioning_Type partition; // object that defines the parallel distribution
partition.SpecifyInternalGhostBoundaryWidths(1,1);

realDistributedArray u(100,100,partition); // build a distributed array
Range I(1,98), J(1,98);

// Parallel array operation with automatic communication:
u(I,J)=.25*( u(I+1,J) + u(I-1,J) + u(I,J+1) + u(I,J-1) ) + sin(u(I,J))/3.;

// Access local serial arrays and call a Fortran routine:
realSerialArray & uLocal = u.getLocalArray(); // access the local array
myFortranRoutine(*uLocal.getDataPointer(),...);
u.updateGhostBoundaries(); // update ghost boundaries on distributed arrays
```

Euler equations: preliminary parallel results



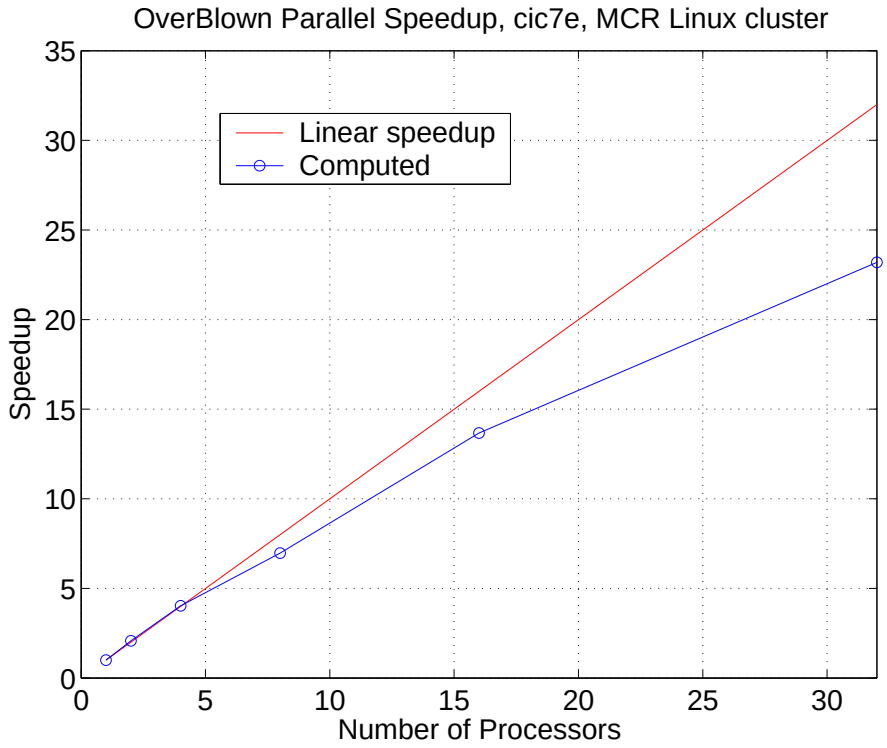
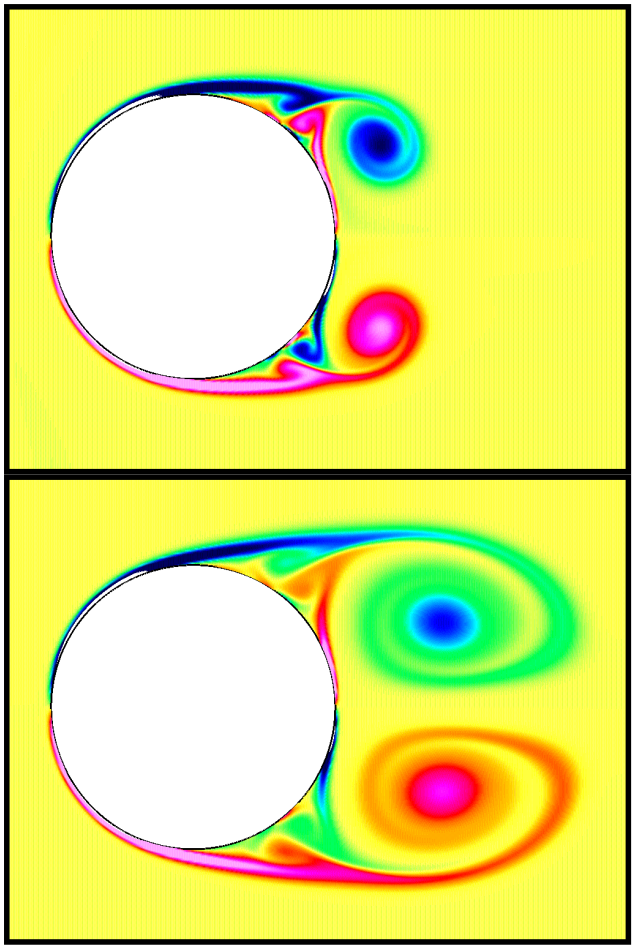


Figure 1: Left: the computation of a shock hitting a cylinder (density). Right: parallel speedup for this problem, keeping the problem size fixed (4 Million grid points), on a linux cluster (Xeon processors).

Incompressible Navier-Stokes: preliminary parallel results



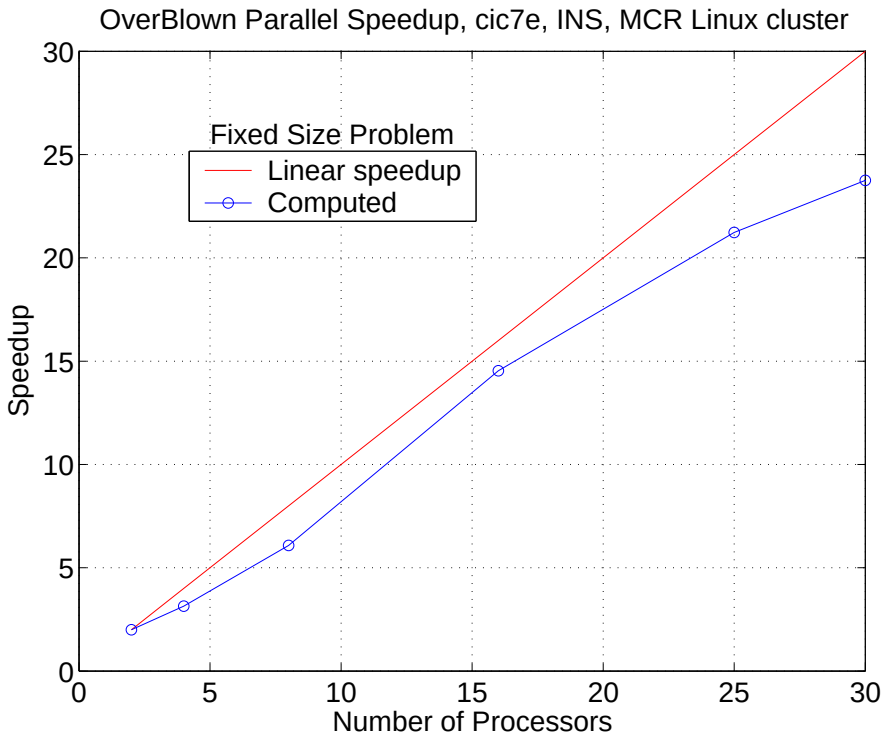


Figure 2: Left: impulsively started cylinder in an incompressible flow (vorticity). Right: parallel speedup keeping the problem size fixed (4 Million grid points), on a linux cluster (Xeon processors). The pressure equation is solved with algebraic multigrid (Hypre).